AFIT/GOA/ENS/98D-01

DEVELOPMENT OF AN OPERATIONS RESEARCH SOFTWARE PACKAGE FOR ARMY DIVISIONS

THESIS

Blane C. Wilson, Captain, USA

AFIT/GOA/ENS/98D-01

AFIT/GOA/ENS/98D-01

DEVELOPMENT OF AN OPERATIONS RESEARCH SOFTWARE PACKAGE FOR ARMY DIVISIONS

THESIS

Presented to the Faculty of the Graduate School of Engineering

Air Force Institute of Technology

Air University

In Partial Fulfillment for the Degree of

Master of Science in Operations Analysis

Blane C. Wilson, B.S.

Captain, USA

December 1998

Approved for public release; distribution unlimited

THESIS APPROVAL

Student: Blane C. Wilson, Captain, USA

Class: GOA-98D

Thesis title: Development of an Operations Research Software Package for Army Divisions

Approved

Committee:	Date
Advisor LTC Jack M. Kloeber, PhD	Dec 10, 1998
Reader Dr. James W. Chrissis, PhD	10 Dec 98

Acknowledgements

I would like to thank my advisor, LTC Jack Kloeber, for his constant guidance and feedback during this research process. Through his vision, he made the idea of this thesis topic a reality. His positive attitude and willingness to provide suggestive feedback kept me motivated until its completion.

I would like to thank my reader, Dr. James Chrissis, for his guidance. His suggestions and comments kept me on course on this thesis.

I would like to thank the Office of the Under Secretary of the Army (Operations Research) for sponsoring this thesis topic.

Lastly, I would like to thank my family, especially my wife Winefreda, for her patience and understanding throughout this thesis research.

Blane C. Wilson

TABLE OF CONTENTS

List of Figures

DEVELOPMENT OF AN OPERATIONS RESEARCH SOFTWARE PACKAGE FOR ARMY DIVISIONS

AFIT/GOA/ENS/98D-01

ABSTRACT

There exists great potential for applying operations research techniques to solve specific problems in the areas of operations, installation support, and training at the Army division level. Because of the operational tempo of today's active-duty and reserve component units, command must focus on accomplishing the daily missions. Also, due their limited knowledge of the field, planners may not be aware of how operations research can be used to enhance planning and operations. Time, training funds, resources, safety, personnel, and equipment are all critical factors in this process. Operations research techniques could be used to improve division-level operations by saving time, managing resources more efficiently, and helping leaders make sound decisions. This thesis research is designed to increase the awareness of how the use of operations research at the division level, can aid planners and decision-makers in solving real problems encountered on a daily basis, thus improving unit operations. By using Microsoft Excel, Visual Basic, and Microsoft Access, a software package was developed to assist division planners in solving problems encountered in such areas as transportation, risk management, fuel service, dining facilities, and shelf storage. Using the software package can result in division planners managing time and resources more effectively.

I. Introduction

Operations Research and Systems Analysis (ORSA) has made a significant impact on United States Army operations since World War II. During this period, Army leaders have addressed many complicated operational problems. Because of the potential loss of lives and equipment, solving problems by experimentation was not feasible. Military leaders applied experience, intuition, imagination, and sometimes the scientific method to solve operational problems. This approach was referred to as "hard core operations ORSA" by Gene Visco, former Director of the U.S. Army Model Improvement and Study Management Agency (MISMA). Visco states that "time and the demand for high level analysis have caused operations ORSA to drift toward analysis that determines facts instead of solving problems."

The role of operations ORSA should return to its original function of providing solutions to operational problems. Walt Hollis, Deputy Under Secretary of the Army (Operations Research) commented that "operations ORSA will get closer to day-to-day Army operations", and he sees the operations ORSA community solving more operational problems. This would include operational problems commanders face daily from "the installation to the battlefield" (David, 1993).

Currently, the United States Army has ten active combat divisions that represent the highest level of tactical command in peacetime and during war. The divisions are located in the United States, Germany, and South Korea. Modern divisions are very large and complex. With the advancement of technology, divisions are integrated with computers which control complex weapons, maintenance, and logistics systems. Soldiers and

commanders are technically proficient in the use of their equipment and tactically proficient in leadership positions. Yet, even with this modern force, the task of solving problems has not decreased in difficulty or importance. In fact, the task has become more complicated because divisions can now deploy anywhere in the world, increasing the planning and logistical requirements.

Typically, a division has 18,000 combat-ready soldiers. With this size, the daily operations conducted by soldiers in a division are enormous, ranging from the very simple to the extremely complicated. For example, on a daily basis, soldiers wait in a line for service; this could be in a dining facility, the central issue facility (CIF), or in an assembly area for fuel. The daily operations might also entail support battalions transporting supplies to combat units in a minimal amount of time, based on the number of available routes. Planners must schedule training and logisticians need to store equipment and supplies in a variety of storage facilities and need to plan for arrival, installation, use, and subsequent breakdown of spare parts.

The daily tasks are accomplished regardless of the level of operations research applied. However, it is possible that a task may not be completed in the most efficient manner thus wasting time, money, or resources on a given task. Many of these operations have had operations research applied to them successfully in industry. In a similar way, the inclusion of operations research to model daily operations should provide beneficial results to units within the division, improving speed, increasing standards, or decreasing resources used.

Operations research gives planners and decision-makers the advantage of making quick, sound decisions through the use of proven techniques. Since operations research

is so effective, why are divisions not using operations ORSA techniques already? Interviewees stated that the use of operations research software was limited due to time, availability for the analysis, training of the users, and accessibility to operations research software. The fundamental objective of this research is to increase the productive use of operations research in the division, by helping planners and decision-makers solve real problems that are encountered on a daily basis through the application of sound operations research methodology.

Problem Statement

There is great potential to utilize OR techniques to solve specific problems in the areas of operations, installation support, and training at the division level. Because of the operational tempo of today's active-duty and reserve component units, leaders are completely focused on successfully accomplishing the daily missions. Due to limited knowledge, planners may not be aware of how operations research can enhance their planning and operations. Time, training funds, resources, safety, personnel, and equipment are all critical factors in this process. Using operations research techniques could improve division-level operations by saving time, managing resources more efficiently, and helping leaders make sound decisions.

Research Objective

To assist the planner and decision-maker, an operations research software package that enables a user to minimize time, minimize cost, manage resources, maximize safety, or optimize large complicated problems can be developed. This can be accomplished by using easy to operate programs, an extensive set of sample problems, and a user's manual.

By the developing this software package for the division, it will give military personnel a better understanding of operations research and the benefits that can be achieved.

Research Scope

The author attempts to solve operational problems that are a representation of problems a division leader encounters in their daily operations. However, this operations research software package will not provide a solution for all of the problems divisions encounter on a daily basis. In addition, the software package includes previously developed analysis software that can be bundled with any original software created in this research. Expectedly, this research generates new ideas and a higher demand for more operational problem solving within the division and therefore requires updates and revisions.

Research Issues

Proprietary software is needed to solve certain problems. In addition, it is important for users to receive training on this operations research software package in order to understand its potential and its limitations. In most cases, a non-operations research analyst can benefit from this product. Finally, future enhancements and modifications for this operations research software package may be needed as the scope and type of operations change in the division and other techniques become available.

Overview

The process in developing the analysis software package is as follows: Chapter 1 defines the problem, defines the research objective, and defines the scope. Chapter 2 discusses the process of finding operations research problems and describes the problems encountered. Chapter 3 discusses the methodology of solving the operational problems.

Chapter 4 provides output results from the operations research software package, and it includes comments from actual users in the division on the operations research software package. Chapter 5 discusses a summary and future research objectives as a result of this analysis package. In addition, it describes the process of writing the users' manual and recording the developed software on a CD-ROM for distribution to all Army divisions.

II. Literature Review

Overview

This chapter discusses some of the problems found in Army divisions and how information about these problems was gathered from personal interviews, phone conversations, e-mail correspondence, historical data from military related web sites, background information on operation research, and U.S. Army manuals. Finally, the chapter concludes with information on operations research software available in divisions.

Visits and Interviews

One method of identifying operational, training, and logistics problems in the division was through personal visits. On April 17, 1998, members of the 101st Air Assault Division General staff at Fort Campbell, Kentucky participated in an interview session designed to identify operational problems. In addition, correspondence with individuals through telephone interviews and attendance at the 1998 Military Operations Research Symposium (MORS) at the Naval Postgraduate School, Monterey, California confirmed previously identified problems and helped point out new problem areas. Listed below are the findings from the interviews and visits.

Problem description: Traffic planning on Army installations

The objective is to analyze existing and projected traffic patterns on Army installations. This analysis would be used to make recommendations that would resolve safety problems that may occur due to changes in traffic volumes on roads and traffic signal intersections. This study would entail a simulation model and a prediction model of a

specific traffic area. The use of this model is specific to the Army installation being studied.

Source: LTC William Klimack, CDR, 2nd Battalion, 28th Infantry Regiment, Fort Jackson, South Carolina

Problem description: Storage optimization problem (external storage)

How can we minimize the storage of different types of equipment in a given area? This leads to questions such as how much equipment can be stored in a cargo shelter on a 3 1/4 ton trailer? How much can be stored in a connex for NTC?

Source: Michael Sullivan, 101st Air Assault Division G4, Fort Campbell, Kentucky

Problem description: Directorate of Public Works (DPW) prioritizes its programs on a constrained budget? With reduced budgets across Army installations, what methods or procedures does the office of DPW use in order to support Army programs such as electrical and water operations, post work order scheduling, post self help agency, fire departments, or construction agencies.

Source: LTC William Klimack, CDR, 2nd Battalion, 28th Infantry Regiment, Fort Jackson, South Carolina

Problem description: Improve range scheduling

Because of limited resources such as training areas, units must schedule and coordinate the use of resources on installations across the Army. How are training resources being used? Is the method of scheduling efficient? Is there a need for a better scheduling system? Is there an effective way to optimize the use of training areas? A study could be performed and techniques developed to optimize the use of training areas.

Source: LTC William Klimack, CDR, 2nd Battalion, 28th Infantry Regiment, Fort Jackson, South Carolina

Problem description: Fuel stations are needed to support vehicles arriving in an assembly area? After an exercise, a battalion has fuel tankers positioned to service supported units to refuel their vehicles. To assist in the planning process of determining the time soldiers spend waiting in line for fuel, this process could be analyzed through a simulation modeled as a multiple server-queue network. The queue discipline is designed to model the behavior of drivers choosing the shortest line, while waiting for fuel. This will allows planners to establish and manage fuel stations, efficiently. This will enable the planner to make a decision on the number of fuel trucks needed as servers? This simulation also could be used in other areas such as evaluating how long soldiers wait in line for service in a dining facility. This might lead to a decision on the number of dining facilities needed to service soldiers. What is amount of time soldiers wait in line to issue items such as MREs, ammunition, fuel, and supplies? This process could be modeled as a simulation with soldiers moving from one station to the next until they have received all of their equipment. How long does it take to issue prescriptions for patients in a medical facility? How long does it take to issue equipment at the central issue facility (CIF)? How long does it take to see patients in a troop medical clinic (TMC) from the time of arrival to the time of departure?

Source: LTC William Klimack, CDR, 2nd Battalion, 28th Infantry Regiment, Fort Jackson, South Carolina

Problem description: How are supply organizations monitoring the requisition of aircraft parts? The DA 1352 document does not pick up all data for reporting purposes.

Is there a way to monitor trends in aircraft parts not recorded on the 1352 documentation in order to evaluate the reliability of a given component?

Source: Michael Sullivan, G4, 101st Air Assault Division

Problem description: Analysis of Hemet across the Army to analyze the most common repair parts in order to evaluate the most reliable components.

Source: Michael Sullivan, G4, 101st Air Assault Division, Fort Campbell, Kentucky

Problem description: A unit commander is required to load a given type of aircraft, such as a C-141. At a moments notice, a smaller transport aircraft, such as a C-130, is assigned as a replacement. The unit commander is directed to load the smaller aircraft, efficiently. Develop a tool that will optimize the movement of supplies via air, based on the available transport aircraft.

Source: Michael Sullivan, G4, 101st Air Assault Division, Fort Campbell, Kentucky

Problem description: Analysis of humanitarian efforts in order to establish documentation of a humanitarian package for peacekeeping operations? What size is the package? How many people to support the package? What material is included in the package?

Source: Timothy Considinet, G5, 101st Air Assault Division, Fort Campbell, Kentucky

Historical data

Another source of operational problems is available at the Center for Army Lessons Learned web page (CALL) located at http://call.army.mil.

Problem description: Supply and distribution of engineer Class IV and V materials to engineer units for defensive operations. Class IV and V resupply for the defense is one of the most demanding logistical operations that a task force (TF) must complete.

Problems documented were the supply and distribution of engineer The result is that units fall short of having required items at a given place and time.

The use of palletized, standardized combat configured loads (CCLs) and the use of the palletized loading system can help resolve the planning and distribution of these materials. The palletized truck has the capability to carry 16.5 tons of supplies, traveling at a maximum speed of 50 mile per hour. The problem of minimizing time across available routes (arcs) to required units (nodes) might be a reliable approach to solve this problem.

Problem description: Risk management is either too time consuming or ignored during the planning process. In some cases, task forces are not identifying and assessing the risk of an operation. An automated risk management tool would assist personnel as they plan and execute a given mission and reduce the time needed for this assessment.

Operations Research Techniques

Using operations research techniques are ideal to solve some of the problems identified in previous sections. Some of the problems identified could be solved as a transportation problem using linear programming, a queuing network using discrete event simulation, a two-dimensional cutting stock problem using dynamic programming, and a risk management worksheet using Microsoft Access.

Transportation problem

Anderson, Sweeney, and Williams (1994) describe how transportation problems are frequently used in planning for the distribution of goods and services from different supply and demand locations. Normally, the quantity of available goods is limited at supply and demand locations. The objective of the transportation problem is to minimize

the total cost of shipping goods from supply to demand locations over available routes. If the supply locations do not have the goods to meet the demand at the destination node, the transportation problem will not have a feasible solution. If a route exists from every available supply node to every available demand node, this is referred to as a complete bipartite graph as stated by (Jensen Notes). If a route is no longer available from a supply node to a demand node, the route is dropped from the network (Anderson, Sweeney, and Williams, 1994) and resolved as a linear program.

Anderson, Sweeney, and Williams, (1994) states that a linear program model can be used to represent the transportation problem. A linear program is an optimization problem that seeks the most or fewest variables in an objective function. The values for the decision variables must satisfy a set of constraints, and there is a sign restriction associated with each variable (Winston, 1994). Linear programming belongs to a deterministic class of problems because the results can be predicted with certainty (Dantzig, 1963). For example, linear programming was used in the food processing industry to determine the routes required to ship ketchup from six plants (supply nodes) to seventy warehouses (demand nodes) (Dantzig, 1963). When solving a transportation problem as a linear program it must have an objective function, supply constraints, and demand constraints (Winston, 1994).

There are a number of linear programming software packages available to solve a transportation problem. The Microsoft Excel spreadsheet is one of the most widely-distributed software packages that includes a built-in solver to optimize linear programs as implemented on a spreadsheet. Today, over 28,000,000 Microsoft Excel users have the optimization solver developed by Frontline Systems, Incorporated (Fylstra, 1998).

The standard software package for the United States Army is Microsoft Office, which includes Microsoft Excel.

Discrete Event Simulations

A discrete event simulation is a model of a system in which state variables only change at discrete points in time. Discrete event simulations can be used to imitate the operation of a real-world process or system over time; they can mimic what happens in real-world systems. The advantages of simulating are that it can provide insight on how a system operates and suggestive improvements for a system under investigation, by changing input and observing the resulting output without actually experimenting on the real system (Banks, Carson, and Nelson, 1996). As such, discrete simulations can be used as a tool for "What if" questions as new systems are designed (Banks, Carson, and Nelson, 1996). However, one major disadvantage is that simulations can be time consuming and difficult to interpret (Banks, Carson, and, Nelson, 1996).

A queuing system could be modeled in a simulation. A queuing system can consist of a number of queues and servers at each queue. Savage (1998) refers to a queuing system with exponentially distributed interarrival and service times with N identical servers as a M/M/N queue. To model a process with more than one queue, can be accomplished by implementing parallel queues in a network. (Savage, 1998). Banks, Carson, and Nelson, (1996) noted that potential customers in this system usually come from a large finite population. Normally, there is a limited amount of space for customers waiting for service in queue (Banks, Carson, and, Nelson, 1996). However, one may assume there is adequate space for customers waiting for service in an assembly area or a dining facility. With a large queue, the importance of a queuing discipline is very important.

A queuing discipline is the logical ordering of customers in a queue, and it determines which customer in a system is chosen for service, as the server becomes free. The queue discipline for customers waiting for service could be the First-in-First-out (FIFO) queuing discipline which means that a customer is serviced in the same order has arrival to the queue (Banks, Carson, and, Nelson, 1996).

The idea of simulating queues has been developed in a number of software packages. It is beneficial to simulate because queues can become too complicated, too quickly, when evaluating analytically. Savage (1998) developed a discrete event simulation for a multiple network queue, using Microsoft Excel spreadsheets. He uses a transition matrix to control the flow of customers among the different stations in the queuing network. A transition matrix is implemented based on Markov chains. A Markov chain is a discrete-time event stochastic process. As an entity moves from one state during one period to another state, this is referred to as a transition. There is a transition probability for Markov chains, which can be represented as a matrix (Winston, 1994). It is very important that all row entries in a transition matrix are nonnegative and each row in the matrix must sum to unity (Winston, 1994). A terminating simulation runs for some duration of time, which is set by the user, and it may have initial conditions such as empty queues or service lines (Banks, Carson, and, Nelson, 1996).

Two-Dimensional Cutting Stock Problem

A two-dimensional cutting stock problem is used to minimize the waste of supplies such as sheets of glass or wood required while satisfying customer orders. The customer's order could require a number of patterns of different lengths and widths. The objective of the cutting stock problem is to minimize the number of sheets required but still meeting the customer's order. For example, a cutting stock problem which customers demanded boards of 40 different lengths involving over 100 million possible ways a board could be cut would very cumbersome to price out the different variables to enter the basis (Winston, 1994). Anderson, Sweeney, and Williams, (1994) concludes that dynamic programming is useful in solving a problem of this type because it allows a large problem to be decomposed into smaller problems that are easier to solve. Winston (1994) points out that by using a column generation approach in solving the cutting stock method, it eliminates the need to price out every nonbasic variable to enter the basis when there are a large number of decision variable. He states the column generation method determines the nonbasic variable that prices out favorably by solving a subproblem referred to as a knapsack problem. By finding the number of items, each of which has a different weight and value, that can be placed in a knapsack with limited weight capacity to maximize the total value of the items placed in the knapsack is referred to as a knapsack problem (Anderson, Sweeney, and, Williams, 1994). Chang-Gon, Myung-Kee, and Hien-Taek coded the two-dimensional cutting stock problem by using dynamic programming and the column generation method, using the BASIC programming language.

Risk Management

FM 100-14 states that risk management "is the process of identifying and controlling hazards to conserve combat power and resources." The steps involved in risk management are identifying the hazard, assessing the hazard, developing controls, assessing residual risk, implementing controls, and supervising and evaluating the problem. A hazard is defined as "any real or potential condition that can cause injury, illness, or death of personnel, or damage to or loss of equipment or property, or mission degradation" (Risk Management, Force XXI presentation slides). After identifying the hazard, the next step is to assess the hazard. A leader must assess the risk degree of the hazard. The risk degree could be <u>low</u> (L), with very little impact on the mission, <u>moderate</u> (M) with an expected degradation of the mission, <u>high</u> (H) with a significant degradation of the mission, or <u>extremely high</u> (EH) with the loss of ability to accomplish the mission. Once the hazards have been assessed, the leader must develop control measures that can be taken to eliminate or reduce risk to hazard (Risk management, Force XXI presentation slides).The next step is to determine residual degree of the risk, which could be <u>low</u> (L), <u>moderate</u> (M), <u>high</u> (H), or <u>extremely high</u> (EH). The next step in the risk management process is to implement control measures by integrating them into standing operating procedures (SOP), written and verbal orders, and mission briefings. Finally, leaders must supervise mission rehearsals to ensure standards and control measures are enforced, and they must continue to monitor the control measures during the mission (FM 100-14). All of the information is recorded on a risk management worksheet as shown in Figure 1.

SAMPLE

SOLUTION - SCENARIO 2

RISK MANAGEMENT WORKSHEET

PAGE 1 of 2

1. MSN/TASK : Perform an Attack by Fire 17-237-10-MTP, (17-3-0219)	**2. DTG BEGIN :** 271500MARXX **END :** 281800MARXX	**3. DATE PREPARED:** 27 MAR XX
4. PREPARED BY: 2 Lt Washington Plt Ldr 2nd Plt B Co 2 /66th Armor RANK/LAST NAME/DUTY POSITION		

5. HAZARDS	6. INITIAL RISK LEVEL	7. CONTROLS	8. RESIDUAL RISK LEVEL	11. HOW TO IMPLEMENT	12. HOW TO SUPERVISE	13. CONTROLS EFFECTIVE
Obstacles - Enemy Tank Ditch - Enemy Wire Obstacle	E	Look for By Pass Route Request Engineer Support for Breeching Operation Actions at an Obstacle Drills	H	Unit TACSOP, FM 17-15 OPORD Rehersal	Cont Supervision Cont Supervision Cont Supervision	
Enemy Forces - 1 Chemical Attack Company - Chemical Agents - Maneuvering in a Chemical Environment	E	SP in MOPP II Mount M8 Alarm on Vehicle Use Tank NBC System when Alarm goes off Perform PMCS on Tank NBC System	H	OPORD Unit TACSOP Unit TACSOP, FM 17-15 TM 9-2350-264-10	Cont Supervision Cont Supervision Cont Supervision Cont Supervision	
Unreliable Soldier Discipline - Uniform - PMCS	H	Brief Leaders and Soldiers on proper Uniform Brief Leaders and Soldiers on importance of PMCS and Discipline	M	Verbal Unit TACSOP, Verbal	Cont Supervision Cont Supervision	
Friendly Fire - Converging Forces - Friendly Artillery Fire and Smoke	E	Graphic Control Measures, Situational Awareness Positive ID Communication , Cross-Talk	H	OPORD ,Unit TACSOP Unit TACSOP Spot Report Situation Report,	Cont Supervision Cont Supervision Verbal Cont Supervision	
Adverse Environmental Conditions - Extreme Heat	H	Implement Hydration Plan Monitor Diet Establish Rest Plan	M	OPORD, TACSOP Unit TACSOP Verbal Instruction	Monitor Spot Check Monitor	

9. OVERALL RISK LEVEL AFTER CONTROLS ARE IMPLEMENTED (CIRCLE ONE): **LOW MODERATE HIGH EXTREMELY HIGH**	10. RISK DECISION AUTHORITY: CPT Greenwood Cdr B Co 2/66th Armor **RANK/LAST NAME/DUTY POSITION**

Figure 1, Risk Management Worksheet (Risk Management, Force XXI Presentation Slides)

Microsoft Office Integrated Software Programs

Microsoft Office 97 has included an object model entitled Data Access Object (DAO) in its software programs. By including DAO as an object in Microsoft Office software packages such as Microsoft Excel 97, a spreadsheet program, and Microsoft Access 97, a database program, it gives programmers the capability of sharing information between the two programs by using Microsoft Visual Basic programming language (Wells and Harshbarger, 1997). By having this capability, users can record data on a spreadsheet

and retrieve additional information from a database, while continuing to work from the spreadsheet.

Military Related Software

Currently, there are analysis software packages available for division units. Some division units use a software package entitled OPLOGPLN '98 to solve logistical requirements. OPLOGPLN '98 is a computer-based program designed to assist logistics planners in calculating supply usage estimates in support of operations. It allows the logistician to calculate supply estimates by class of supply, specifically supply classes I, II, III (Bulk and Packaged), IV, V (Conventional and Bulk), VI, VII, VIII, and water. OPLOGPLN '98 is designed specifically to support operations typically associated with multi-phase operation plans (OPLAN) and operation orders (*OPORD).* This software enables personnel to develop operations orders as they forecast different requirements. OPLOGPLN '98 requires an IBM-compatible PC with an Intel 80386 processor or higher, 4 MB or more of RAM and at least 45 MB of free hard drive space (30 MB for installation, 10 MB for swap files and at least 5 MB for user files). OPLOGPLN '98 is a DOS-based program and will run under MS-DOS 3.3 or higher (MS-DOS 5.0 or higher recommended). It will also run as a DOS program under Windows 3.1, Windows 95 and Windows NT 4.0. OPLOGPLN '98 is distributed by CASCOM. A request for OPLOGPLN '98 can be made via the Internet (Combined Arms Support Command Home Page, http://www.cascom.army.mil/multi/operations_logistics_planner).

Another software package available in the division is the Military Application Program Package (MAPP). This software was developed in 1989; it is a DOS menu driven program that includes operational research software. MAPP provides users with

the capability of solving linear programming problems with 21 decision variables and 21 constraints. MAPP also includes linear regression analysis, movement planning, personnel status, and a decision matrix. MAPP requires IBM-compatible PC with an Intel 80286 processor or higher. The complete installation requires 7 KB of data space. The MAPP package assumes the user is familiar with operations research. MAPP is not distributed from any official military organization.

III. Methodology

Overview

This chapter discusses the approach taken to solve some of the problems identified during the research process. The problems discussed are: the transportation problem, risk management problem, fuel-point queuing network, dining facility wait line, and the shelf storage problem. The identified problems were solved using Microsoft Excel, Microsoft Access, and Microsoft Visual Basic. The objective of this chapter is to present the methods and software used in solving the identified problems.

Transportation Problem

A division consists of three maneuver brigades. Each maneuver brigade is assigned a forward support battalion (FSB). It is organized with a headquarters and headquarters detachment, a supply company, a maintenance company with designated system support teams, and a medical company. The FSB provides dedicated support to the same brigade in garrison and in tactical operations. Once deployed to the field, the FSB also provides area support to divisional elements operating in the brigade sector as well as brigade supporting elements such as artillery, air defense, and engineer assets (FM 71-123, Combat Service Support). While on a field exercise, a FSB may want to minimize the time required to deliver wood (Class IV), fuel (Class V), or spare parts (Class IV) to subordinate units. What routes should be selected? How many trips are necessary? This problem could be solved as a linear program by minimizing on the overall time of transporting class IV, class V, or IX supplies from supply nodes (FSB) to demand nodes (support units) across available routes. The transportation problem is designed as a ten-node supply and ten-node demand network

system where available routes may exist between different supply and demand nodes. The input parameters to the model will include route mileage, supply numbers, demand numbers, available vehicles at each supply node, vehicle load capacity, speed movement across routes, and units assigned to each supply and demand node. In addition, the design of the program has a feature that allows the user to put an upper and lower bound constraint on supply nodes (FSB). The design provides a visual representation of the available routes from supply nodes (FSB) to demand nodes (support units). In addition, the visual representation shows the routes selected, after the linear program model is solved.

The linear program (LP) is implemented using Microsoft Excel and solved by accessing the add-in solver through Microsoft Visual Basic. The result provides the planner with valuable information such as the time to deliver supplies on a route based on speed movement, the number of transport vehicles used over the optimal routes, the number trips required on the route, and the load amount delivered over the optimal routes.

Risk Management Problem

Human error, equipment malfunction, or environmental conditions can result in accidents. For fiscal year 1998, there were a total of 2,144 Army accidents, with a total of 171 fatalities as a result. In addition the total accident cost was $182 million dollars (U.S. Army Safety Center Web Page, http://safety.army.mil). The lesson learned is that leaders must continue to include risk management in garrison and on training exercises in order to better protect the military force.

To assist planners and decision-makers in using risk management, an automated risk management worksheet would be a useful tool. This software reduces the amount of time to complete a risk management worksheet. In addition, the automated risk management

worksheet gives the decision-maker alternatives in interpreting overall risks for a given mission. The automated risk management worksheet is similar to the worksheet shown in Figure 1, Chapter 2. The automated worksheet is designed to allow users to retrieve requested information from a database file to an Excel worksheet through the database access object (DAO). If the information is not currently available in the database, the risk management program is designed to allow the user to update the database with new information by running a Visual Basic form. The Visual Basic form allows as a user to input data directly to a Microsoft Access database, without opening the Microsoft Access program. By linking a database to the risk management worksheet, this enables the planner to retrieve stored information rapidly on future generated worksheets.

Because different leaders may have different interpretations of overall risk, this worksheet is designed using three ways to calculate the overall risk level after control levels are implemented. The first approach in calculating the overall risk factor is for decision-makers that are concerned with the overall average of all risk factors for a given mission. The overall risk level is based on a <u>low</u> (L), <u>moderate</u> (M), <u>high</u> (H), or <u>extremely</u> <u>high</u> (EH) degree of risk for a given task. The overall risk is measured from an assigned baseline probability, which can be modified. Each risk factor is assigned a probability value. <u>Low</u> has a probability value of 0, <u>moderate</u> has a probability value of 0.3, <u>high</u> has a probability value of 0.7, and <u>extremely</u> <u>high</u> has a probability of 1. The risk factors are summed and averaged on the worksheet. The overall risk factor is determined by finding the minimal difference between the baseline probability values and the value of the averaged risk factors. For example, on a risk worksheet with two <u>low</u> (L) risk factors and three <u>moderate</u> (M) risk

factors assigned, the average is 0.18. Because the minimal distance is closer to the moderate (M) baseline value, the overall risk level is moderate (M).

The second approach in determining the overall risk level is for decision-makers to determine the mode of the risk factors for a given mission. In a given set of values, the mode is the most frequently occurring value. For example, on a risk worksheet with two low (L) risk factors and three moderate (M) risk factors assigned, the mode is moderate (M). Therefore, the overall risk level is assigned the value of moderate (M).

Finally, the third approach in determining the overall risk is for decision-makers that are concerned with the highest risk factor for a given mission. For example, on a risk worksheet with two low (L) risk factors, three moderate (M) risk factors, and one high (H) risk assigned, The overall risk level is high (H). A more conservative decision-maker would use this overall risk calculation. The conservative approach is the one apparently recommended by FM 100-14.

Fuel Service Problem

Long delays of vehicles in an assembly could cause logistical and scheduling problems. In addition, there is the safety issue if too many vehicles are overcrowded into one area. This problem is designed to assist planners in estimating how much time drivers spend waiting in line for service, the length of the fuel line, or the number of fuel operators required in a designated area such as an assembly area. Each queue is based on a first-in, first-out (FIFO) queuing discipline. The queue priority is assigned to the last fuel service position by the use of *IF* statements in Microsoft Excel. This queue priority is needed in order to break ties between fuel service positions with equal queue lengths. The arrival rate is based on a uniform distribution, and the service rate is based on a Poisson distribution. The user inputs

the arrival time of the drivers, the mean service time of the fuel service positions, the total simulation run time, and the number of service positions available. The user may select from one to seven fuel service positions. Twenty-five replications are conducted before the output results are displayed. The output results are based on the assigned priority queue. The output results provide the planner with the average wait time a single driver is expected to wait in line for fuel and a confidence interval on the average wait time.

The confidence interval provides information on the distribution of the twenty-five replications. Based on a normal distribution, the planner is provided with a 50% and a 95% confidence interval about the true mean. In addition, a 95% prediction interval, which provides the planner with information on the average wait time a driver, is expected to wait in line for service in the future. In addition, a histogram of the average waiting time is provided for the priority queue. The average waiting time is based on the priority queue. The output provides insight to the staff officer on the overall fuel operation in an assembly. Therefore, it gives the staff officer the opportunity to readjust his planning prior to executing a fuel operation. Based on the output, the staff officer may add or delete a fuel service position. The staff officer could adjust the arrival of vehicles to the assembly area, or he could adjust the service time for fuel service positions, especially if the results signify a long delay for drivers in the assembly area. For this simulation, the staff officer is able to acquire additional fuel operators, and the staff officer has sufficient fuel on hand. Finally, the staff officer has planned for an assembly area sufficient for drivers to wait in line for fuel service.

Dining Facility Problem

The problem is to model how long soldiers wait in line in a dining facility for service. The soldiers move through the line from server to server through the dining facility. The output results are determined by summing a soldier's wait time for service at each server position in line. The output results display a histogram for the average waiting time for soldiers and the total number of soldiers serviced in a dining facility. Based on the output results, the decision-maker could cease serving a particular food for a given period, add more servers, adjust dining facility hours, or shut down a dining facility. For example, if it is essential that a Brigade move through a dining facility quickly, the decision-maker might eliminate serving a specialty dish such as an omelet, which may take up to three minutes to serve. Instead, scrambled eggs are prepared, which take only seconds to serve. It is assumed that sufficient food is available at each service position and sufficient space for soldiers to wait in line for food. Because the queuing discipline is based on FIFO, it is assumed that soldiers cannot skip a service position while waiting in line.

Waiting time in a troop medical clinic (TMC)

The problem is to model how long soldiers or dependents wait to see medical personnel. It is assumed that the waiting area is adequate to hold all patients. Based on the output results, the decision-maker could require more medical personnel such as doctors, nurses, or physician-assistants to service patients.

Optimization storage problem

The objective is to minimize the total number of shelves required to store a given set of equipment. A solution to this type of problem provides the decision-maker with the number of shelves required storing equipment on a shelf, and it provides the number of items stored on each shelf. The solution does not provide detailed information on the exact placement of an item on a shelf.

This optimization problem is used to determine how to store different types of equipment in a given area of space. It is assumed that the equipment is in rectangular boxes. The model minimizes the number of shelves required to store boxes of varied dimensions. In addition, the model provides the user with the types of boxes stored on each shelf. The user will input the length, width, and height of the different types of boxes. Using visual basic macros, the boxes will be rearranged to the smallest two-dimensional "foot print" before the data is optimized. In addition, the maximal height separation is found for the shelves. This design uses the two-dimensional cutting stock code developed by Chang-on, Myung-Kee, and Hien-Taek, (1995) using the delayed column generation method. The code was augmented with Microsoft Excel and Visual Basic.

IV. Results

Overview

This chapter shows the output results from the software developed for the transportation problem, fuel service problem, dining facility problem, risk management worksheet, and optimization storage problem. The problems were solved by using Microsoft Excel, Microsoft Access, and Visual Basic. The designed software was evaluated by military personnel from the 3rd Infantry, Division, Fort Stewart, Georgia from 22 September – 25 September 1998. The military personnel provided useful comments about the designed software. Overall, the author received positive feedback on the software programs.

Transportation Problem

The transportation problem consists of a maximum ten-supply and ten-demand node network. The user controls the input parameters for the network by activating available macro buttons and input cells. First, the user initiates the routes available from the distribution centers (supply nodes) to the combat units (demand nodes) by clicking on the macro buttons, and the number of miles for each route as shown in Figure 2.

ROUTES	MILES	ROUTES	MILES	ROUTES	MILES	ROUTES	MILES	ROUTES	MILES
	1		32		4		2		444
	3		11		2		1		234
	3		56		111		5		567
	4		90		23		6		128
	5		11		1		7		10
	6		777		19		8		12
	111		777		1		9		14
	113		67		32		10		16
	12		55		3		11		17
	47		34		2		12		18

Figure 2 Routes and Miles Input Sheet (partial snapshot)

Based on the routes selected from the input sheet, the transportation network illustrates the available routes as shown in Figure 3. The left side of the network represents the supply nodes and the right side of the network represents the demand nodes.

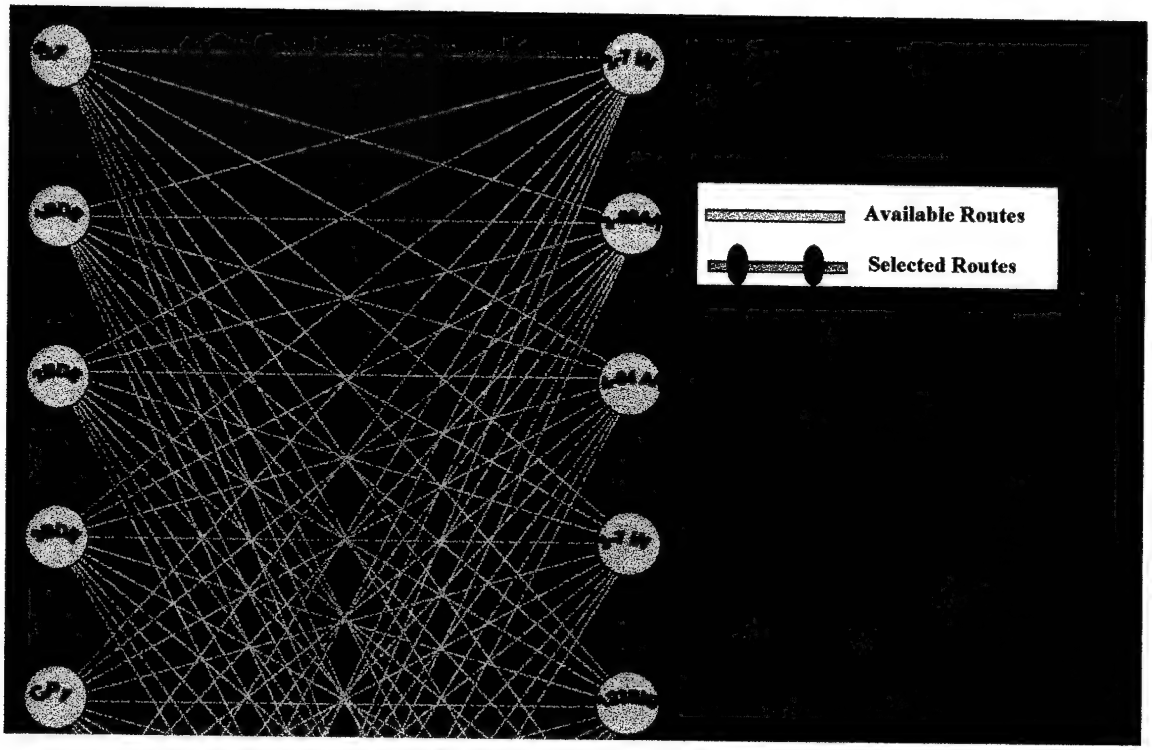


Figure 3 Transportation Network (partial snapshot)

The name of the unit, the amount of available supplies (supply nodes), the amount of demanded supplies (demand nodes), the rate of travel of vehicles over routes, the number of vehicles available at each supply node, and the capacity of vehicles are entered on the input worksheet as shown in Figure 4.

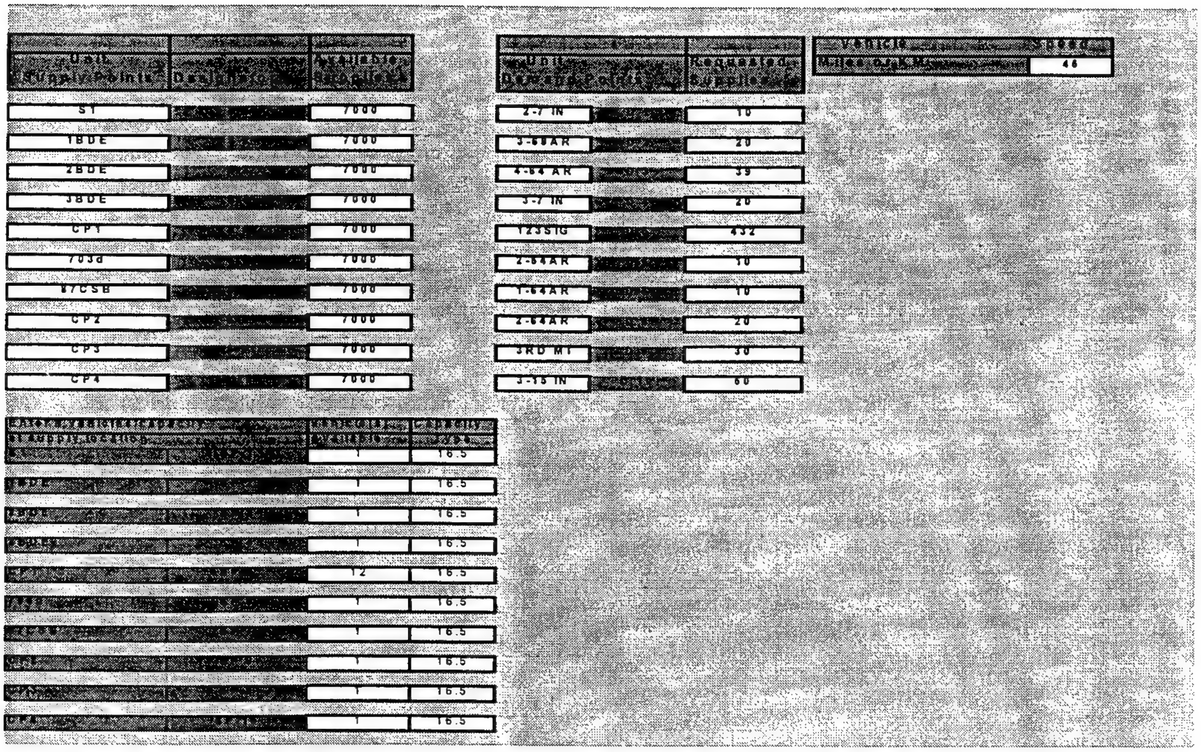


Figure 4 Supply, Demand, Vehicle, and Capacity Input Sheet

Once optimized, the optimal routes for delivering supplies to demand nodes are shown in Figure 5. The total time (in minutes) to deliver equipment from a supply node to a demand node, the total load required at a demand node, the capacity sum of vehicles transporting goods to a demand node, the total deliveries, and the mileage from a supply node to a demand node is shown in Figure 6. For example, the first line of the table shows supply node S1 will deliver 10 tons of supplies to 2-7 IN in 1.3 minutes. It will take one trip and one transport vehicle (16.5 tons) to meet the demands for 2-7 IN.

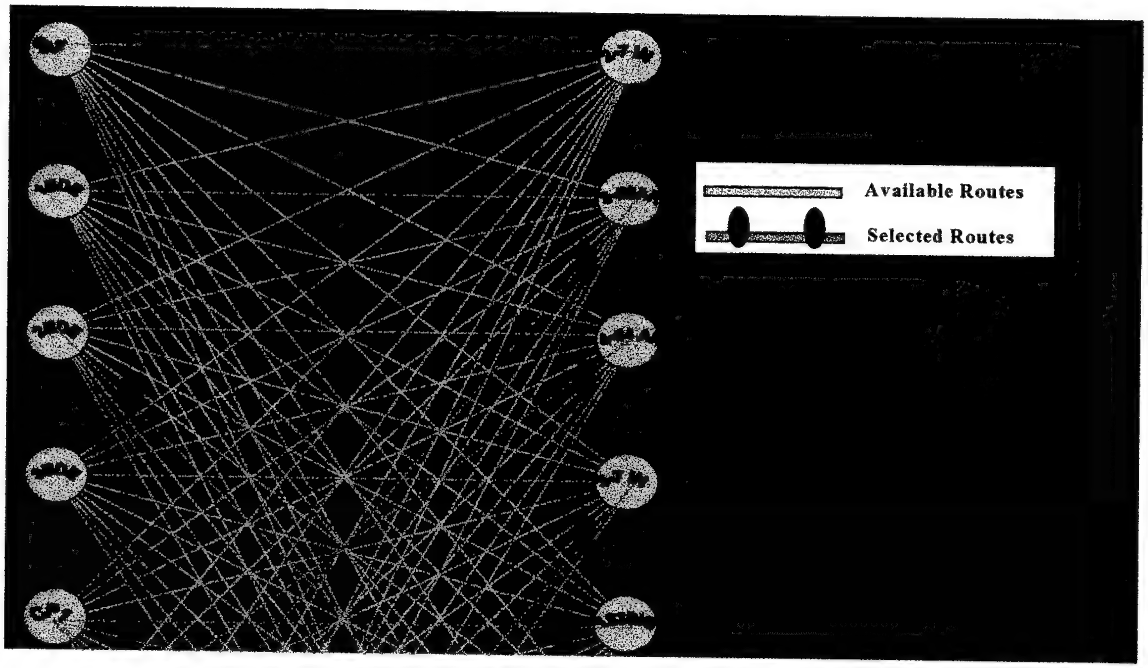


Figure 5 Optimal Routes Selected

Total Route (min)	Time In Delivery Supplies to I			Load Amount	Capacity n of available vehi	Per Route Deliveries	Miles on Route
1.3	S1	TO	2-7 IN	10	16.5	1	1
0	S1	TO	3-69AR	0	16.5	0	3
0	S1	TO	4-64 AR	0	16.5	0	3
0	S1	TO	3-7 IN	0	16.5	0	4
0	S1	TO	123SIG	0	16.5	0	5
0	S1	TO	2-54AR	0	16.5	0	6
0	S1	TO	1-64AR	0	16.5	0	111
0	S1	TO	2-64AR	0	16.5	0	113
0	S1	TO	3RD MI	0	16.5	0	12
0	S1	TO	3-15 IN	0	16.5	0	47
0	1BDE	TO	2-7 IN	0	16.5	0	32
0	1BDE	TO	3-69AR	0	16.5	0	11
0	1BDE	TO	4-64 AR	0	16.5	0	56
0	1BDE	TO	3-7 IN	0	16.5	0	90
0	1BDE	TO	123SIG	0	16.5	0	11
0	1BDE	TO	2-54AR	0	16.5	0	777
0	1BDE	TO	1-64AR	0	16.5	0	777
0	1BDE	TO	2-64AR	0	16.5	0	67
0	1BDE	TO	3RD MI	0	16.5	0	55
0	1BDE	TO	3-15 IN	0	16.5	0	34
0	2BDE	TO	2-7 IN	0	16.5	0	4
0	2BDE	TO	3-69AR	0	16.5	0	2
0	2BDE	TO	4-64 AR	0	16.5	0	111
0	2BDE	TO	3-7 IN	0	16.5	0	23
0	2BDE	TO	123SIG	0	16.5	0	1
0	2BDE	TO	2-54AR	0	16.5	0	19
1.3	2BDE	TO	1-64AR	10	16.5	1	1

Figure 6 Output Table

. After evaluating the information in the table, the user may want a specific supply node to deliver an amount of goods. This option is available by changing the parameters as noted in Figure 7. The parameters show a minimum and maximum requirement for each supply node. Once these parameters are changed, the problem must be resolved.

(Minimum Requirement Supply Usage	
S1	0
1BDE	0
2BDE	0
3BDE	0
CP1	0
703d	0
87CSB	0
CP2	0
CP3	0
CP4	0

(Maximum Requirement) Supply Usage	
S1	100000
1BDE	100000
2BDE	100000
3BDE	100000
CP1	100000
703d	100000
87CSB	99999
CP2	100000
CP3	100000
CP4	100000

Figure 7 Supply Usage Options

CW2 Ramsey, a supply system technician, assigned to the HHC 24th CSG, stated that his organization could use the program. He states the program is useful for finding the best resupply points to deliver goods to demand nodes. He states the program is excellent for transportation or fuel operations in a tactical environment. In addition, he states the program is good for military operations requiring the distribution of supplies to combat units.

MAJ Roach, a G-4 Division plans/operations officer, assigned to 3rd Infantry Division stated that the program would be very useful for units moving large quantities of supplies over different routes. He states the program is ideal for a transportation unit. In addition, he stated this problem could be used to transport bulk fuel from supply to demand nodes.

Fuel Service Problem

The software enables a planner to manage the operations of a fueling operation in an assembly area. The planner inputs the mean arrival time for fuel, service time for each fuel position, and the total simulation time as shown in Figure 8. Currently, the worksheet shows four fuel service positions available, with each position having a service time of five minutes. In addition, it shows a arrives every two minutes on average. The total simulation time for this problem is one hundred and twenty minutes.

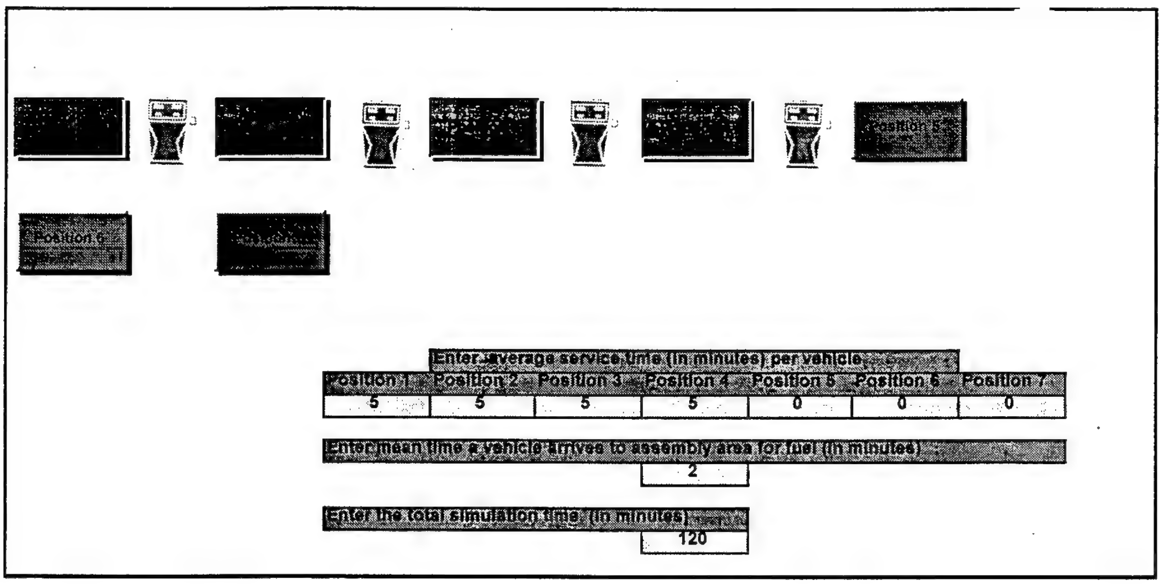
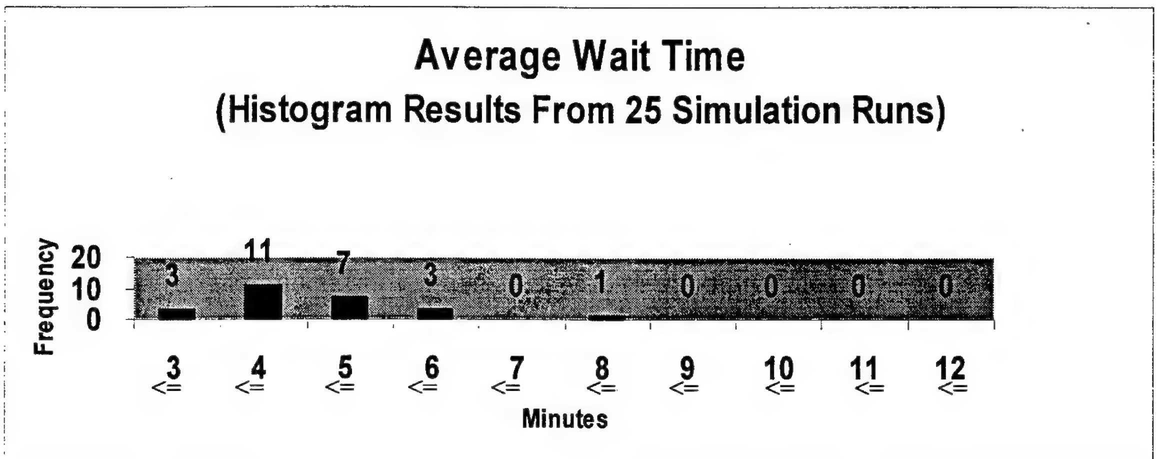


Figure 8 Fuel Service Input

After running the simulation for twenty-five replications, the output provides the decision-maker with valuable information about the system being modeled: the sample mean (average wait time in queue) a driver is expected to wait in line for fuel, a standard deviation of the sample mean (average wait time in queue), a 50% and 95% confidence interval of driver's waiting time, and a 95% prediction confidence interval, if this fuel operation with the same input parameters were repeated in the future. It also provides the

decision-maker with information on the minimum value and maximum value of the sample mean (average wait time in queue) of the simulation run. The histogram from the twenty-five replications is shown in Figure 9. For example, 11 of 25 runs are less than or equal to 4 minutes but greater than 3 minutes. The minimum wait time for each driver is 2.43 minutes, and the maximum wait time is 7.09 minutes. The mean (average wait time in the queue) is 4.08 minutes. The 95% confidence interval is 3.65 minutes to 4.51 minutes.



Max Value	7.09	minutes	Min Value	2.43	minutes	minutes
Mean(Average Wait Time) a driver waits in line for service					4.08	minutes
Variance (Average Wait Time) a driver waits in line before receiving service					1.08	minutes
Standard Deviation (Average Wait Time) a driver waits in line before receiving service					1.04	minutes
3.94	minutes to	4.22	minutes is a 50% confidence interval			
3.65	minutes to	4.51	minutes is a 95% confidence interval			
1.89	minutes to	6.27	minutes is a 95% prediction interval a driver expects to wait for service in the future			

Figure 9 Fuel Service Histogram (Wait Time)

CW2 Ramsey suggested that the fuel service program is appropriate for the 24th CSG since they have a fuel and water battalion. Also, CPT Doug Henry, 2nd Brigade Assistant S-4, 3rd Infantry Division stated the fuel service program could be a valuable tool in determining the proper assets required in establishing a Refuel On the Move (ROM) for Army ground operations.. In addition, he stated movement control officers (MCO) within a logistics section could utilize this program to adjust march unit intervals to avoid an extensive wait time at a ROM assembly area.

CPT John Hinson, 1st Brigade Assistant S-4, 3rd Infantry Division was impressed with the fuel program because it could be utilized by the planner within the division to effectively manage resources. He stated the fuel program is more useful in a main or forward support battalion because they have more fuel assets assigned to their organization.

Dining Facility Problem

The dining facility problem provides insight on the operations of a dining facility. The input parameters are shown in Figure 10 is the sheet used to input the arrival time for soldiers to a dining facility, service time for each server position, and the total simulation time. Currently, the worksheet shows four dining facility servers with a soldier arriving on average every 12 seconds, and a service time of 12 seconds for each service position. The total simulation time for the model is 60 minutes.

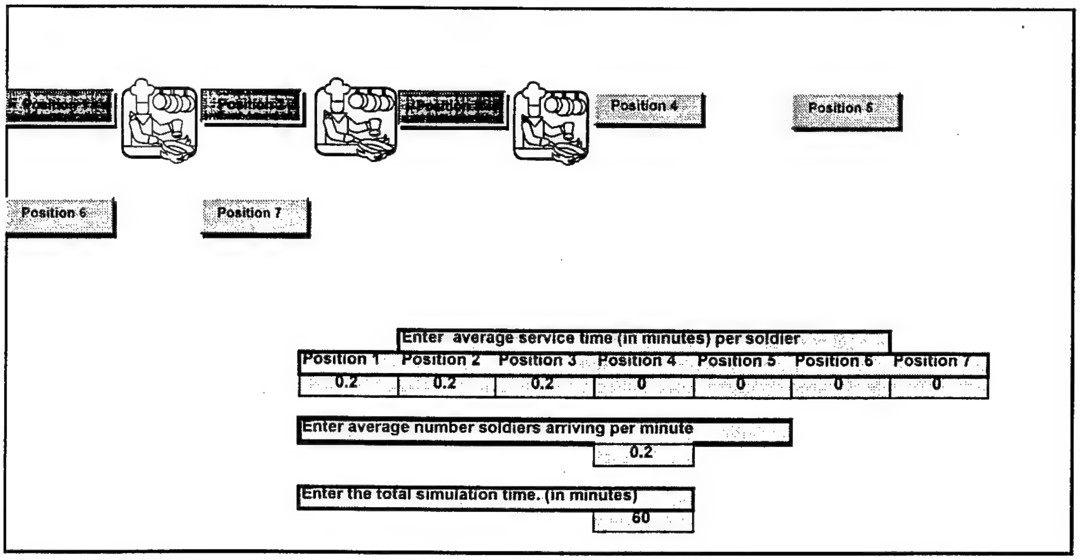
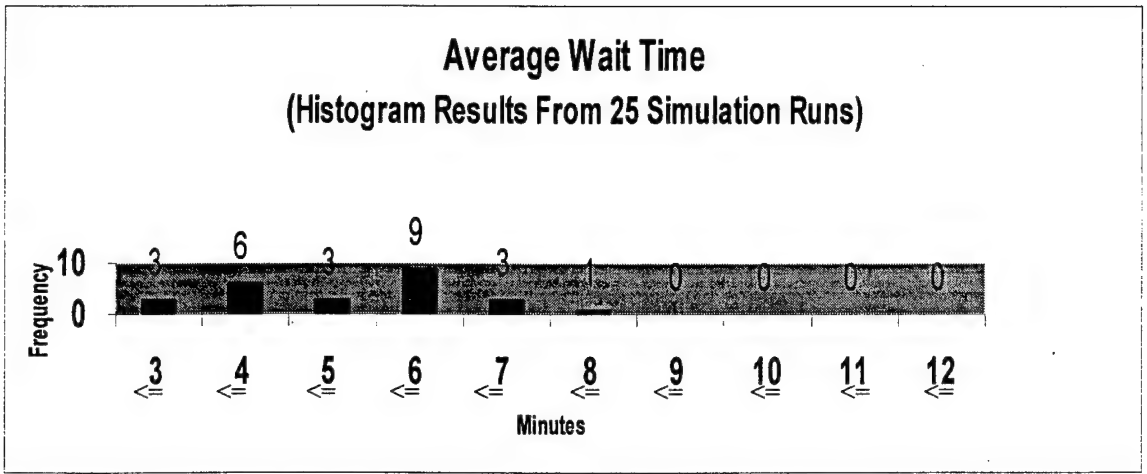


Figure 10 Dining Facility Input

After running the simulation for twenty-five replications, the output provides the decision-maker with valuable information about the system being modeled. The output result provides the decision-maker with the mean (average wait time in queue) a soldier is expected to wait in line for food. It provides the decision-maker with a standard deviation of the mean (average wait time in queue). It provides the decision-maker with a 50% and 95% confidence interval for the time a soldier waits in line for food. Finally, it provides the decision-maker with information on the minimum value and maximum value of the simulation run.

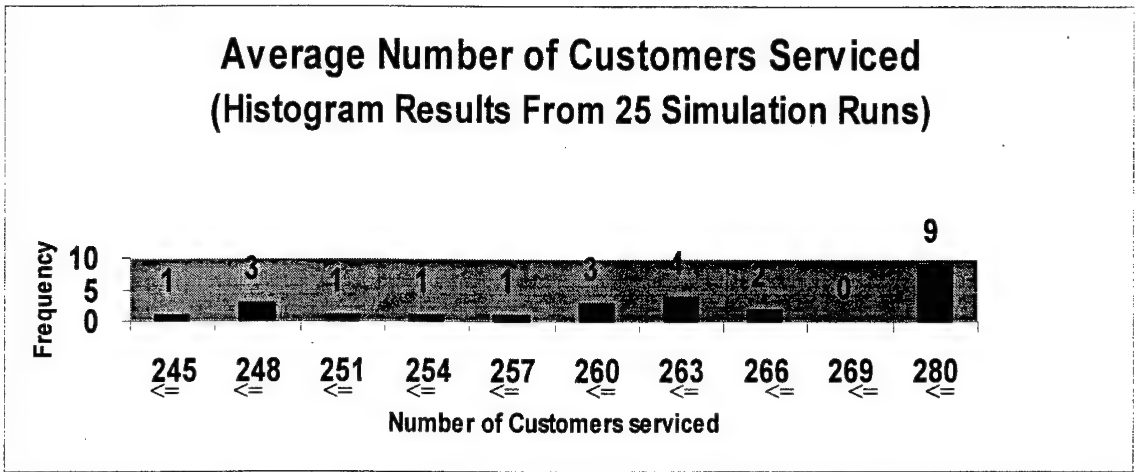
Figure 11 shows the histogram from the twenty-five replications. For example, 3 of 25 runs show the average wait time for soldier's waiting in line for food is less than 3 minutes. The minimum wait time from the twenty-five replications is 2.46 minutes, and the maximum wait time is 9.6 minutes. The mean (average wait time in the queue) is 5.75 minutes. Finally, the 95% confidence interval of the wait time falls between 4.12 minutes and 5.3 minutes.



Max Value	7.92	minutes	Min Valu	2.37	minutes	minutes
Sample Mean(Average Wait Time) a driver waits in line for service					4.71	minutes
Variance (Average Wait Time) a driver waits in line before receiving service					2.07	minutes
Standard Deviation (Average Wait Time) a driver waits in line before receiving					1.44	minutes
4.42	minutes to	5	minutes is a 50% confidence interval			
4.12	minutes to	5.3	minutes is a 95% confidence interval			
3.24	minutes to	6.18	minutes is a 95% prediction interval a driver expects to wait for service in the future			

Figure 11 Dining Facility Histogram (Wait Time)

Additional information about the number of customers serviced is shown in Figure 12. For example, 1 of 25 runs show the average number of soldiers serviced in the dining facility is less than or equal to 245 soldiers. The minimum number serviced is 245 soldiers, and the maximum number serviced is 280 soldiers. The mean (number of customer serviced) is 262 soldiers. Finally, a 95% confidence interval shows the number serviced fall between 258 soldiers and 267 soldiers. This program was not evaluated during the visit to Fort Stewart, Georgia. However, the features are similar to the fuel program.



Max Value	280	customers	Min Valu	245	customers	customers
Mean(Number of customers serviced)					262.68	customers
Variance (Number of customers serviced)					109.98	customers
Standard Deviation (Number of customers serviced)					10.49	customers
261.24	customers	264.12	customers	is a 50% confidence interval		
258.35	customers	267.01	customers	is a 95% confidence interval		
240.61	customers	284.75	customers	is a 95% prediction confidence interval (future)		

Figure 12 Dining Facility Histogram (Serviced)

Automated Risk Management Sheet

A blank automated risk management worksheet is shown in Figure 13. The features on this worksheet are similar to the risk management sheet discussed in Chapter II under risk management. The additional features on this worksheet are the macro buttons, which are used to initialize, clear, and calculate the risk factors on the worksheet.

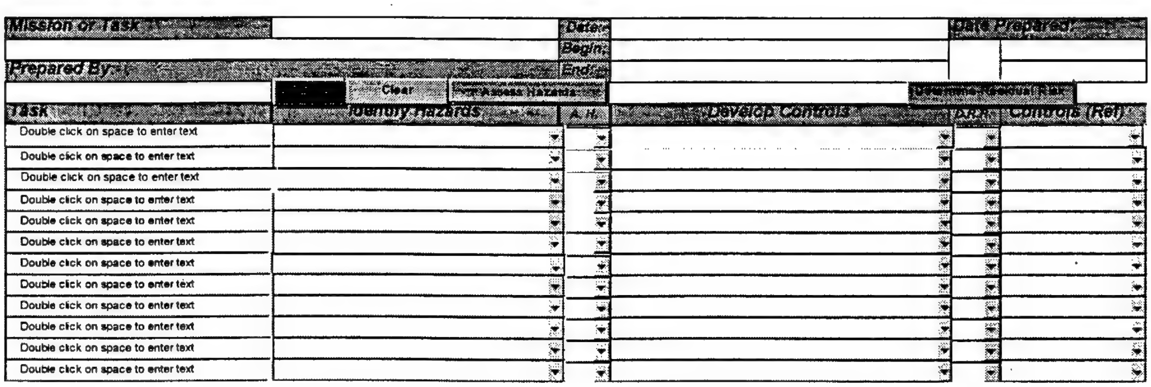
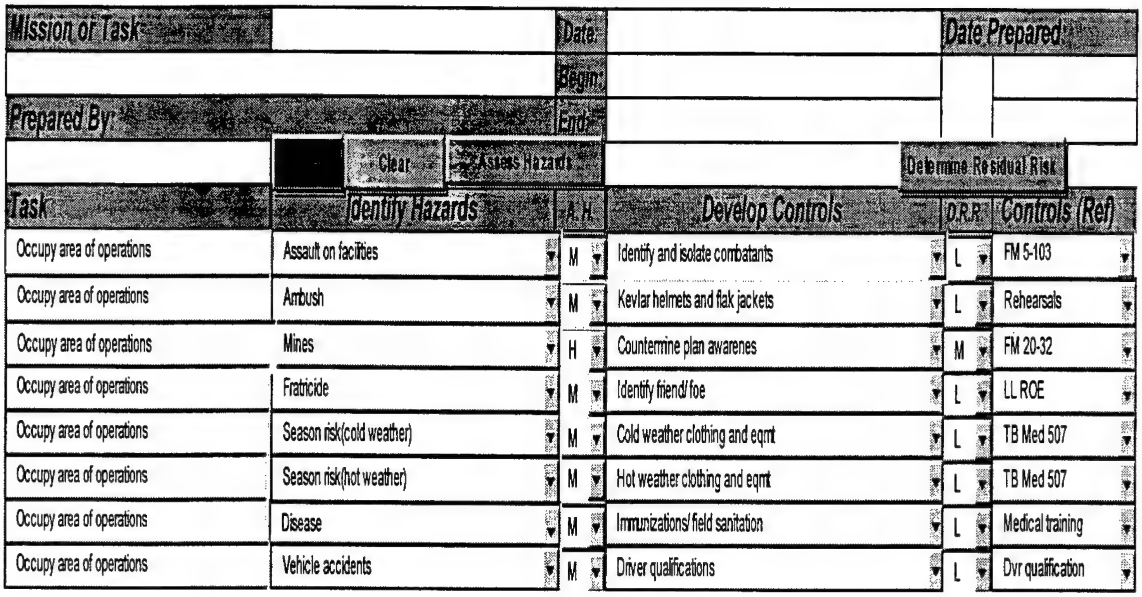


Figure 13 Risk Management Worksheet (blank)

Figure 14 shows the risk management worksheet with information.



Mission or Task: | Date: | Begin: | End: | Date Prepared:

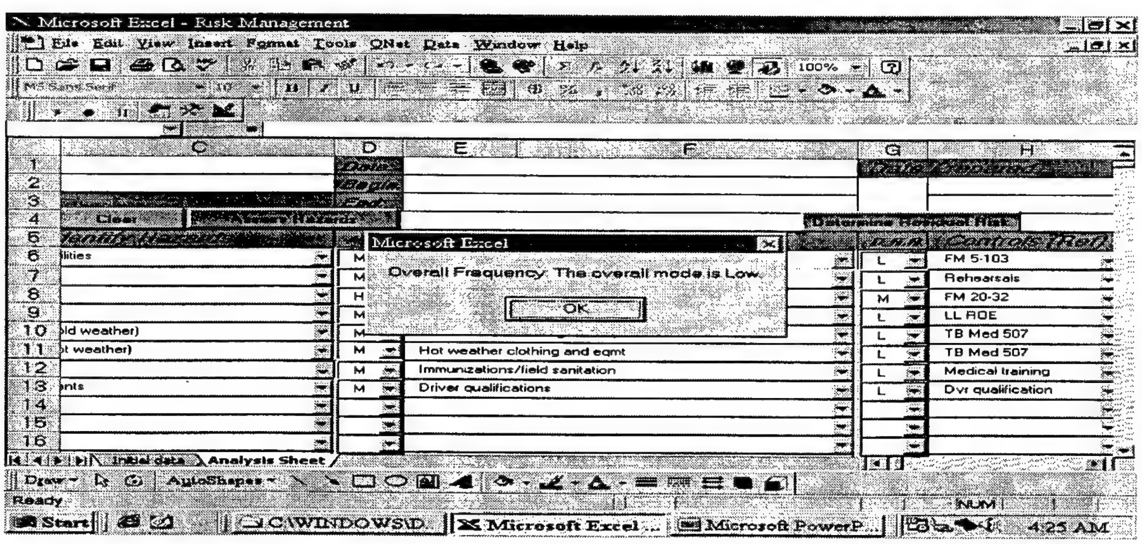
Prepared By:

Clear | Assess Hazards | Determine Residual Risk

Task	Identify Hazards	A.H.	Develop Controls	D.R.R.	Controls (Ref)
Occupy area of operations	Assault on facilities	M	Identify and isolate combatants	L	FM 5-103
Occupy area of operations	Ambush	M	Kevlar helmets and flak jackets	L	Rehearsals
Occupy area of operations	Mines	H	Countermine plan awarenes	M	FM 20-32
Occupy area of operations	Fratricide	M	Identify friend/ foe	L	LL ROE
Occupy area of operations	Season risk(cold weather)	M	Cold weather clothing and eqmt	L	TB Med 507
Occupy area of operations	Season risk(hot weather)	M	Hot weather clothing and eqmt	L	TB Med 507
Occupy area of operations	Disease	M	Immunizations/ field sanitation	L	Medical training
Occupy area of operations	Vehicle accidents	M	Driver qualifications	L	Dvr qualification

Figure 14 Risk Management Worksheet (with data)

Once the information is complete, the overall risk factors are calculated as described in the methodology in Chapter III. The output for each overall risk factor is shown in Figure 15, Figure 16, and Figure 17.



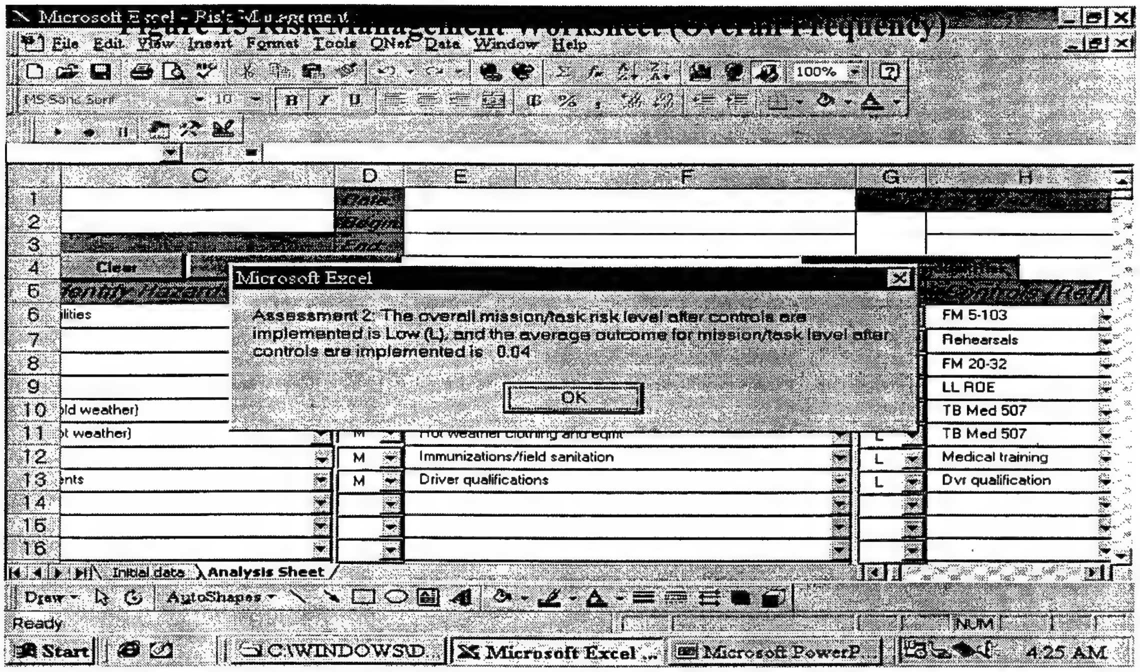


Figure 16 Risk Management Worksheet (Average)

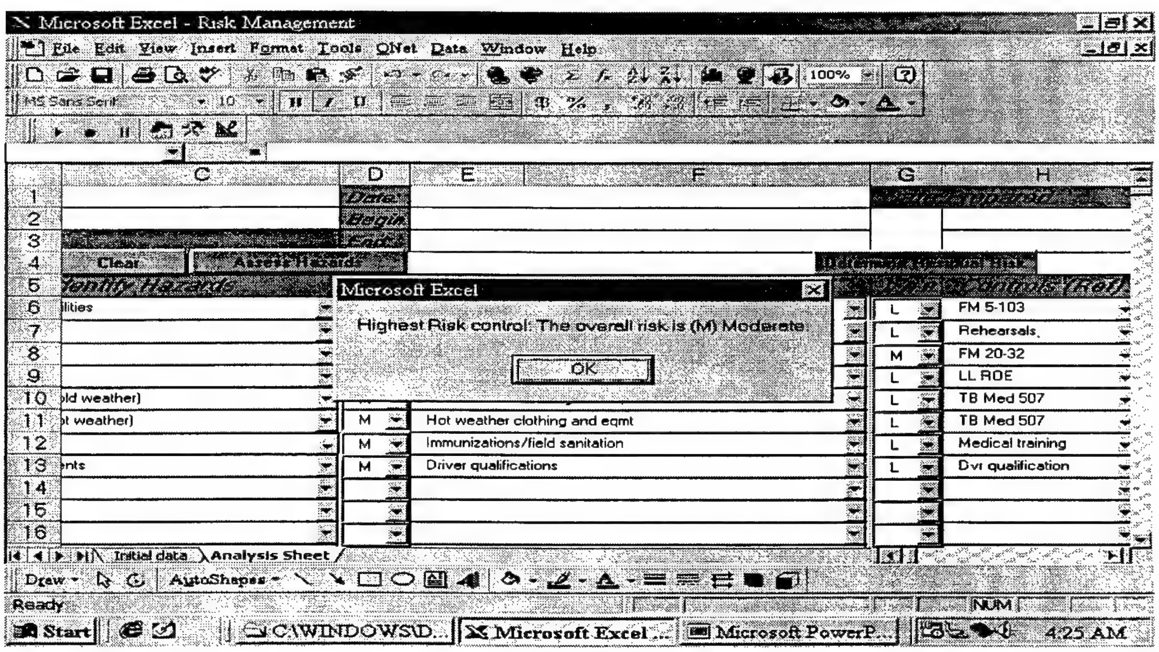


Figure 17 Risk Management Worksheet (Highest)

CPT Phillip McDonald, Assistant S-3, 123d Signal Battalion, stated the risk management program was very easy to use and comparable to the manual from FM 100-

14. He suggested the program could be used as a supplement in a risk management course. He noted the only draw back to the program was the dependency of a computer to run the program.

Optimization Storage Problem

The input sheet for the storage problem is shown in Figure 18. The input sheet requires an item name, item length, item width, item height, item quantity, storage length, and storage width. For example, item 1 has a length of 5 and a width of 5. The quantity demanded is 20 for item 1. Once the sheet is calculated, the maximum height of a storage shelf is displayed. In this example, the shelf height is 5.

Item Name	Item Length	Item Width	Item Height	Item Quantity
1	5	5	5	20
2	4	1	2	20
3	3	1	1	20
4	2	1	1	20
5	7	6	2	20
6	8	1	3	20
7	9	3	4	20
8	10	1	4	20

5
10
10
100
8

Figure 18 Storage Problem

In addition, the result gives information on which items are stored on each shelf and the number of shelves required. For example, the worksheet shows item 6, with an 8 x 1 dimension requires shelf space for twenty boxes. The pattern(s) with 8 x 1 are highlighted. Pattern 3 shows ten 8 x 1 dimensions will fit on 1 shelf. Pattern 4 shows

one 8 x 1 dimension will fit on 1 shelf. Pattern 5 shows one 8 x 1 dimension will fit on each of the fifteen shelves for this pattern. Because of rounding, the results may overestimate the requirement, slightly. In this example, there was a requirement to place twenty 8 x 1 items on shelves. The program found a solution to place twenty-six, 8 x 1 items on shelves. Each storage shelf is of a 10 x 10 dimension. Twenty-eight storage shelves are needed to meet the total requirement for all items as noted in Figure 18.

```
**** Optimal configuration Pattern ****

Pattern ( 1 )
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 5.
Maximum number of item(s) with a 5 * 5  dimension per storage shelf is 4 .

Pattern ( 2 )
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 2.
Maximum number of item(s) with a 4 * 1 dimension per storage shelf is 8.
Maximum number of item(s) with a 3 * 1 dimension per storage shelf is 6.
Maximum number of item(s) with a 7 * 6 dimension per storage shelf is 1.

Pattern ( 3 )
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 1.
Maximum number of item(s) with a 2 * 1 dimension per storage shelf is 10.
Maximum number of item(s) with an 8 * 1 dimension per storage shelf is 10.

Pattern ( 4 )
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 1.
Maximum number of item(s) with a 3 * 1 dimension per storage shelf is 6.
Maximum number of item(s) with a 2 * 1 dimension per storage shelf is 1.
Maximum number of item(s) with a 7 * 6 dimension per storage shelf is 1.
Maximum number of item(s) with an 8 * 1 dimension per storage shelf is 1.
Maximum number of item(s) with a 9 * 3 dimension per storage shelf is 1.

Pattern ( 5 )
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 15.
Maximum number of item(s) with a 7 * 6 dimension per storage shelf is 1.
Maximum number of item(s) with an 8 * 1 dimension per storage shelf is 1.
Maximum number of item(s) with a 9 * 3 dimension per storage shelf is 1.
```

Pattern (6)
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 1.
Maximum number of item(s) with a 4 * 1 dimension per storage shelf is 2.
Maximum number of item(s) with a 2 * 1 dimension per storage shelf is 7.
Maximum number of item(s) with a 7 * 6 dimension per storage shelf is 1.
Maximum number of item(s) with a 9 * 3 dimension per storage shelf is 1.

Pattern (7)
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 1.
Maximum number of item(s) with a 4 * 1 dimension per storage shelf is 2.
Maximum number of item(s) with a 9 * 3 dimension per storage shelf is 3.

Pattern (8)
Storage rectangle with a 10 * 10 dimension.
The number of storage shelves required for this pattern is 2.
Maximum number of item(s) with a 10 * 1 dimension per storage shelf is 10.

No more than 28 shelves are required.

V. Conclusion

Summary

During this research process, a number of operational problems were encountered in Army divisions. After researching the problems, the appropriate operations research techniques were used to solve the problems discussed during this research.

The problems solved were the transportation problem, fuel service problem, dining facility problem, risk management worksheet, and optimization problem. By allowing military personnel to evaluate the programs, it exposed them to the benefits of using operations research in their daily operations, and it allowed personnel to provide feedback on the usefulness of the developed programs. However, to fine-tune any additional changes in the software, more trials should be conducted in other Army divisions.

By using Microsoft Visual Basic, Microsoft Excel, and Microsoft Access, the distribution of the software is very feasible and cost effective, since a majority of the Army units have Microsoft Office products. Copies of the operations research software package should be sent directly to the division G-3, operations section and the division G-4, logistics section. The sections can distribute the copies to logistic support battalions and battalion operations sections within the division.

Recommendations

The idea of solving problems for divisions should continue for future research projects. There is a variety of unsolved operational problems such as traffic planning on Army installations, range scheduling, or budget management issues. This software package is a valuable tool for decision-makers and planners as they plan and execute

daily missions. As an enhancement to this research, the operations research software could be moved to different software platform such as java programming, which could be integrated on a web page.

VI. Visual Basic Code

Appendix A: Transportation Problem

This routine activates the user form under Visual Basic Application for Microsoft Excel.

```
Sub Distro()
Distribution.Show
End Sub
```

The following six subroutines are the button selections appearing on the user form when activated.

This subroutine makes the supply and demand worksheet the activate sheet under Microsoft Excel. After activating sheet, the user form selections is unloaded.

```
Private Sub CommandButton7_Click()
SupplyD
Unload Distribution
End Sub
```

This subroutine makes the option worksheet the activate sheet under Microsoft Excel. After activating sheet, the user form selections is unloaded.

```
Private Sub CommandButton13_Click()
Options
Unload Distribution
End Sub
```

This subroutine makes the table worksheet the activate sheet under Microsoft Excel. After activating sheet, the user form selections is unloaded.

```
Private Sub CommandButton6_Click()
Table
Unload Distribution
End Sub
```

This subroutine makes the route worksheet the activate sheet under Microsoft Excel. After activating sheet, the user form selections is unloaded.

```
OptimalRoutes
Unload Distribution
End Sub
```

This routine updates any changes made under the options, S&D, and input worksheet, before making a call to the solve routine.

```
Private Sub CommandButton4_Click()
max
min
```

```
miles
SandP
For I = 5 To 104
Worksheets("Engine").Cells(I, 1) = 0
Next I
Solve
Unload Distribution
End Sub
```

This subroutine activates the Net subroutine. After updating the sheet, the user form selections is unloaded.

```
Private Sub CommandButton3_Click()
Net
Unload Distribution
End Sub
```

This subroutine initializes solver and loads the transportation problem, based on the settings calculated. After loading the problem, the user form is unloaded.

```
Private Sub CommandButton8_Click()
SolverReset
Worksheets("Engine").Select

init = 5
x = 5
y = 1
While Cells(x, y) <> ""
x = x + 1
'MsgBox "Numbers: " & x
Wend

SolverOK SetCell:=Range("Time"), _
MaxMinVal:=2, _
ByChange:=Range(Cells(init, y), Cells(x - 1, y))

SolverAdd CellRef:=Range("$A$5:$A$104"), _
Relation:=3, _
FormulaText:=0

SolverAdd CellRef:=Range("$I$112:$R$112"), _
Relation:=1, _
FormulaText:="$I$114:$R$114"

SolverAdd CellRef:=Range("$S$112:$AB$112"), _
Relation:=2, _
```

Appendix A: Transportation Problem

```
FormulaText:="$S$114:$AB$114"



SolverAdd CellRef:=Range("$A$231"), _
Relation:=3, _
FormulaText:=("$A$232")
SolverAdd CellRef:=Range("$B$231"), _
Relation:=3, _
FormulaText:=("$B$232")
SolverAdd CellRef:=Range("$C$231"), _
Relation:=3, _
FormulaText:=("$C$232")
SolverAdd CellRef:=Range("$D$231"), _
Relation:=3, _
FormulaText:=("$D$232")
SolverAdd CellRef:=Range("$E$231"), _
Relation:=3, _
FormulaText:=("$E$232")
SolverAdd CellRef:=Range("$F$231"), _
Relation:=3, _
FormulaText:=("$F$232")
SolverAdd CellRef:=Range("$G$231"), _
Relation:=3, _
FormulaText:=("$G$232")
SolverAdd CellRef:=Range("$H$231"), _
Relation:=3, _
FormulaText:=("$H$232")
SolverAdd CellRef:=Range("$I$231"), _
Relation:=3, _
FormulaText:=("$I$232")
SolverAdd CellRef:=Range("$J$231"), _
Relation:=3, _
FormulaText:=("$J$232")
SolverAdd CellRef:=Range("$A$231"), _
Relation:=1, _
FormulaText:=("$A$233")
SolverAdd CellRef:=Range("$B$231"), _
Relation:=1, _
FormulaText:=("$B$233")
SolverAdd CellRef:=Range("$C$231"), _
Relation:=1, _
FormulaText:=("$C$233")
SolverAdd CellRef:=Range("$D$231"), _
Relation:=1, _
FormulaText:=("$D$233")
```

Appendix A: Transportation Problem

```
SolverAdd CellRef:=Range("$E$231"), _
Relation:=1, _
FormulaText:=("$E$233")
SolverAdd CellRef:=Range("$F$231"), _
Relation:=1, _
FormulaText:=("$F$233")
SolverAdd CellRef:=Range("$G$231"), _
Relation:=1, _
FormulaText:=("$G$233")
SolverAdd CellRef:=Range("$H$231"), _
Relation:=1, _
FormulaText:=("$H$233")
SolverAdd CellRef:=Range("$I$231"), _
Relation:=1, _
FormulaText:=("$I$233")
SolverAdd CellRef:=Range("$J$231"), _
Relation:=1, _
FormulaText:=("$J$233")

SolverOptions AssumeLinear:=True
SolverOptions Derivatives:=1
SolverOptions SearchOption:=1
SolverOptions Estimates:=1
SolverOptions MaxTime:=32767
SolverOptions Iterations:=32767

SolverOptions Precision:=0.0000001, _
Convergence:=0.001
Unload Distribution
End Sub
```

This routine selects the worksheets("Options") as the activate sheet.

```
Sub Options()
Worksheets("Options").Select
End Sub
```

This routine selects the worksheets("SD") as the activate sheet.

```
Sub SupplyD()
Worksheets("SD").Select
End Sub
```

This routine copies the solved results from one worksheet to another.

Appendix A: Transportation Problem

```
Sub Table()
Sheets("Engine").Select
Range("A112:D213").Select
Selection.Copy
Sheets("Solved Table").Select
Range("A1").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
False, Transpose:=False
Range("$A$2").Select
End Sub
```

This routines makes the worksheet "Route" the active worksheet and activates the routine legend.

```
Sub OptimalRoutes()
Worksheets("Routes").Select
legend
Range("A2").Select
End Sub
```

This routine adds a legend to the routes worksheet.

```
Sub legend()
Sheets("Engine").Select
ActiveSheet.Shapes("Group 29").Select
Selection.Copy
Sheets("Routes").Select
Range("H8").Select
ActiveSheet.Paste
End Sub
```

This routine updates the upper bound values selected under the options worksheet to the worksheet engine.

```
Sub max()
Worksheets("Engine").Cells(233, 1) = Worksheets("Options").Cells(3, 9)
Worksheets("Engine").Cells(233, 2) = Worksheets("Options").Cells(4, 9)
Worksheets("Engine").Cells(233, 3) = Worksheets("Options").Cells(5, 9)
Worksheets("Engine").Cells(233, 4) = Worksheets("Options").Cells(6, 9)
Worksheets("Engine").Cells(233, 5) = Worksheets("Options").Cells(7, 9)
Worksheets("Engine").Cells(233, 6) = Worksheets("Options").Cells(8, 9)
Worksheets("Engine").Cells(233, 7) = Worksheets("Options").Cells(9, 9)
Worksheets("Engine").Cells(233, 8) = Worksheets("Options").Cells(10, 9)
Worksheets("Engine").Cells(233, 9) = Worksheets("Options").Cells(11, 9)
Worksheets("Engine").Cells(233, 10) = Worksheets("Options").Cells(12, 9)
```

End Sub

This routine updates the lower bound values selected under the options worksheet to the worksheet engine.

```
Sub min()
Worksheets("Engine").Cells(232, 1) = Worksheets("Options").Cells(3, 3)
Worksheets("Engine").Cells(232, 2) = Worksheets("Options").Cells(4, 3)
Worksheets("Engine").Cells(232, 3) = Worksheets("Options").Cells(5, 3)
Worksheets("Engine").Cells(232, 4) = Worksheets("Options").Cells(6, 3)
Worksheets("Engine").Cells(232, 5) = Worksheets("Options").Cells(7, 3)
Worksheets("Engine").Cells(232, 6) = Worksheets("Options").Cells(8, 3)
Worksheets("Engine").Cells(232, 7) = Worksheets("Options").Cells(9, 3)
Worksheets("Engine").Cells(232, 8) = Worksheets("Options").Cells(10, 3)
Worksheets("Engine").Cells(232, 9) = Worksheets("Options").Cells(11, 3)
Worksheets("Engine").Cells(232, 10) = Worksheets("Options").Cells(12, 3)
End Sub
```

This routine updates miles values selected under the input worksheet to the worksheet engine.

```
Sub miles()
Worksheets("Engine").Cells(5, 4) = Worksheets("Input").Cells(2, 3)
Worksheets("Engine").Cells(6, 4) = Worksheets("Input").Cells(4, 3)
Worksheets("Engine").Cells(7, 4) = Worksheets("Input").Cells(6, 3)
Worksheets("Engine").Cells(8, 4) = Worksheets("Input").Cells(8, 3)
Worksheets("Engine").Cells(9, 4) = Worksheets("Input").Cells(10, 3)
Worksheets("Engine").Cells(10, 4) = Worksheets("Input").Cells(12, 3)
Worksheets("Engine").Cells(11, 4) = Worksheets("Input").Cells(14, 3)
Worksheets("Engine").Cells(12, 4) = Worksheets("Input").Cells(16, 3)
Worksheets("Engine").Cells(13, 4) = Worksheets("Input").Cells(18, 3)
Worksheets("Engine").Cells(14, 4) = Worksheets("Input").Cells(20, 3)
Worksheets("Engine").Cells(15, 4) = Worksheets("Input").Cells(2, 7)
Worksheets("Engine").Cells(16, 4) = Worksheets("Input").Cells(4, 7)
Worksheets("Engine").Cells(17, 4) = Worksheets("Input").Cells(6, 7)
Worksheets("Engine").Cells(18, 4) = Worksheets("Input").Cells(8, 7)
Worksheets("Engine").Cells(19, 4) = Worksheets("Input").Cells(10, 7)
Worksheets("Engine").Cells(20, 4) = Worksheets("Input").Cells(12, 7)
Worksheets("Engine").Cells(21, 4) = Worksheets("Input").Cells(14, 7)
Worksheets("Engine").Cells(22, 4) = Worksheets("Input").Cells(16, 7)
Worksheets("Engine").Cells(23, 4) = Worksheets("Input").Cells(18, 7)
Worksheets("Engine").Cells(24, 4) = Worksheets("Input").Cells(20, 7)
Worksheets("Engine").Cells(25, 4) = Worksheets("Input").Cells(2, 10)
Worksheets("Engine").Cells(26, 4) = Worksheets("Input").Cells(4, 10)
Worksheets("Engine").Cells(27, 4) = Worksheets("Input").Cells(6, 10)
Worksheets("Engine").Cells(28, 4) = Worksheets("Input").Cells(8, 10)
```

Appendix A: Transportation Problem

```
Worksheets("Engine").Cells(29, 4) = Worksheets("Input").Cells(10, 10)
Worksheets("Engine").Cells(30, 4) = Worksheets("Input").Cells(12, 10)
Worksheets("Engine").Cells(31, 4) = Worksheets("Input").Cells(14, 10)
Worksheets("Engine").Cells(32, 4) = Worksheets("Input").Cells(16, 10)
Worksheets("Engine").Cells(33, 4) = Worksheets("Input").Cells(18, 10)
Worksheets("Engine").Cells(34, 4) = Worksheets("Input").Cells(20, 10)
Worksheets("Engine").Cells(35, 4) = Worksheets("Input").Cells(2, 13)
Worksheets("Engine").Cells(36, 4) = Worksheets("Input").Cells(4, 13)
Worksheets("Engine").Cells(37, 4) = Worksheets("Input").Cells(6, 13)
Worksheets("Engine").Cells(38, 4) = Worksheets("Input").Cells(8, 13)
Worksheets("Engine").Cells(39, 4) = Worksheets("Input").Cells(10, 13)
Worksheets("Engine").Cells(40, 4) = Worksheets("Input").Cells(12, 13)
Worksheets("Engine").Cells(41, 4) = Worksheets("Input").Cells(14, 13)
Worksheets("Engine").Cells(42, 4) = Worksheets("Input").Cells(16, 13)
Worksheets("Engine").Cells(43, 4) = Worksheets("Input").Cells(18, 13)
Worksheets("Engine").Cells(44, 4) = Worksheets("Input").Cells(20, 13)
Worksheets("Engine").Cells(45, 4) = Worksheets("Input").Cells(2, 16)
Worksheets("Engine").Cells(46, 4) = Worksheets("Input").Cells(4, 16)
Worksheets("Engine").Cells(47, 4) = Worksheets("Input").Cells(6, 16)
Worksheets("Engine").Cells(48, 4) = Worksheets("Input").Cells(8, 16)
Worksheets("Engine").Cells(49, 4) = Worksheets("Input").Cells(10, 16)
Worksheets("Engine").Cells(50, 4) = Worksheets("Input").Cells(12, 16)
Worksheets("Engine").Cells(51, 4) = Worksheets("Input").Cells(14, 16)
Worksheets("Engine").Cells(52, 4) = Worksheets("Input").Cells(16, 16)
Worksheets("Engine").Cells(53, 4) = Worksheets("Input").Cells(18, 16)
Worksheets("Engine").Cells(54, 4) = Worksheets("Input").Cells(20, 16)
Worksheets("Engine").Cells(55, 4) = Worksheets("Input").Cells(23, 3)
Worksheets("Engine").Cells(56, 4) = Worksheets("Input").Cells(25, 3)
Worksheets("Engine").Cells(57, 4) = Worksheets("Input").Cells(27, 3)
Worksheets("Engine").Cells(58, 4) = Worksheets("Input").Cells(29, 3)
Worksheets("Engine").Cells(59, 4) = Worksheets("Input").Cells(31, 3)
Worksheets("Engine").Cells(60, 4) = Worksheets("Input").Cells(33, 3)
Worksheets("Engine").Cells(61, 4) = Worksheets("Input").Cells(35, 3)
Worksheets("Engine").Cells(62, 4) = Worksheets("Input").Cells(37, 3)
Worksheets("Engine").Cells(63, 4) = Worksheets("Input").Cells(39, 3)
Worksheets("Engine").Cells(64, 4) = Worksheets("Input").Cells(41, 3)
Worksheets("Engine").Cells(65, 4) = Worksheets("Input").Cells(23, 7)
Worksheets("Engine").Cells(66, 4) = Worksheets("Input").Cells(25, 7)
Worksheets("Engine").Cells(67, 4) = Worksheets("Input").Cells(27, 7)
Worksheets("Engine").Cells(68, 4) = Worksheets("Input").Cells(29, 7)
Worksheets("Engine").Cells(69, 4) = Worksheets("Input").Cells(31, 7)
Worksheets("Engine").Cells(70, 4) = Worksheets("Input").Cells(33, 7)
Worksheets("Engine").Cells(71, 4) = Worksheets("Input").Cells(35, 7)
Worksheets("Engine").Cells(72, 4) = Worksheets("Input").Cells(37, 7)
Worksheets("Engine").Cells(73, 4) = Worksheets("Input").Cells(39, 7)
Worksheets("Engine").Cells(74, 4) = Worksheets("Input").Cells(41, 7)
```

Appendix A: Transportation Problem

```
Worksheets("Engine").Cells(75, 4) = Worksheets("Input").Cells(23, 10)
Worksheets("Engine").Cells(76, 4) = Worksheets("Input").Cells(25, 10)
Worksheets("Engine").Cells(77, 4) = Worksheets("Input").Cells(27, 10)
Worksheets("Engine").Cells(78, 4) = Worksheets("Input").Cells(29, 10)
Worksheets("Engine").Cells(79, 4) = Worksheets("Input").Cells(31, 10)
Worksheets("Engine").Cells(80, 4) = Worksheets("Input").Cells(33, 10)
Worksheets("Engine").Cells(81, 4) = Worksheets("Input").Cells(35, 10)
Worksheets("Engine").Cells(82, 4) = Worksheets("Input").Cells(37, 10)
Worksheets("Engine").Cells(83, 4) = Worksheets("Input").Cells(39, 10)
Worksheets("Engine").Cells(84, 4) = Worksheets("Input").Cells(41, 10)
Worksheets("Engine").Cells(85, 4) = Worksheets("Input").Cells(23, 13)
Worksheets("Engine").Cells(86, 4) = Worksheets("Input").Cells(25, 13)
Worksheets("Engine").Cells(87, 4) = Worksheets("Input").Cells(27, 13)
Worksheets("Engine").Cells(88, 4) = Worksheets("Input").Cells(29, 13)
Worksheets("Engine").Cells(89, 4) = Worksheets("Input").Cells(31, 13)
Worksheets("Engine").Cells(90, 4) = Worksheets("Input").Cells(33, 13)
Worksheets("Engine").Cells(91, 4) = Worksheets("Input").Cells(35, 13)
Worksheets("Engine").Cells(92, 4) = Worksheets("Input").Cells(37, 13)
Worksheets("Engine").Cells(93, 4) = Worksheets("Input").Cells(39, 13)
Worksheets("Engine").Cells(94, 4) = Worksheets("Input").Cells(41, 13)
Worksheets("Engine").Cells(95, 4) = Worksheets("Input").Cells(23, 16)
Worksheets("Engine").Cells(96, 4) = Worksheets("Input").Cells(25, 16)
Worksheets("Engine").Cells(97, 4) = Worksheets("Input").Cells(27, 16)
Worksheets("Engine").Cells(98, 4) = Worksheets("Input").Cells(29, 16)
Worksheets("Engine").Cells(99, 4) = Worksheets("Input").Cells(31, 16)
Worksheets("Engine").Cells(100, 4) = Worksheets("Input").Cells(33, 16)
Worksheets("Engine").Cells(101, 4) = Worksheets("Input").Cells(35, 16)
Worksheets("Engine").Cells(102, 4) = Worksheets("Input").Cells(37, 16)
Worksheets("Engine").Cells(103, 4) = Worksheets("Input").Cells(39, 16)
Worksheets("Engine").Cells(104, 4) = Worksheets("Input").Cells(41, 16)
End Sub
```

This routine updates the supply and demand values selected under the SD worksheet to the worksheet engine.

```
Sub SandP()
Worksheets("Engine").Cells(114, 9) = Worksheets("SD").Cells(7, 3)
Worksheets("Engine").Cells(114, 10) = Worksheets("SD").Cells(9, 3)
Worksheets("Engine").Cells(114, 11) = Worksheets("SD").Cells(11, 3)
Worksheets("Engine").Cells(114, 12) = Worksheets("SD").Cells(13, 3)
Worksheets("Engine").Cells(114, 13) = Worksheets("SD").Cells(15, 3)
Worksheets("Engine").Cells(114, 14) = Worksheets("SD").Cells(17, 3)
Worksheets("Engine").Cells(114, 15) = Worksheets("SD").Cells(19, 3)
Worksheets("Engine").Cells(114, 16) = Worksheets("SD").Cells(21, 3)
```

Appendix A: Transportation Problem

```
Worksheets("Engine").Cells(114, 17) = Worksheets("SD").Cells(23, 3)
Worksheets("Engine").Cells(114, 18) = Worksheets("SD").Cells(25, 3)
Worksheets("Engine").Cells(114, 19) = Worksheets("SD").Cells(7, 7)
Worksheets("Engine").Cells(114, 20) = Worksheets("SD").Cells(9, 7)
Worksheets("Engine").Cells(114, 21) = Worksheets("SD").Cells(11, 7)
Worksheets("Engine").Cells(114, 22) = Worksheets("SD").Cells(13, 7)
Worksheets("Engine").Cells(114, 23) = Worksheets("SD").Cells(15, 7)
Worksheets("Engine").Cells(114, 24) = Worksheets("SD").Cells(17, 7)
Worksheets("Engine").Cells(114, 25) = Worksheets("SD").Cells(19, 7)
Worksheets("Engine").Cells(114, 26) = Worksheets("SD").Cells(21, 7)
Worksheets("Engine").Cells(114, 27) = Worksheets("SD").Cells(23, 7)
Worksheets("Engine").Cells(114, 28) = Worksheets("SD").Cells(25, 7)
Worksheets("Engine").Cells(5, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(6, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(7, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(8, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(9, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(10, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(11, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(12, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(13, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(14, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(15, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(16, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(17, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(18, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(19, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(20, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(21, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(22, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(23, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(24, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(25, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(26, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(27, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(28, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(29, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(30, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(31, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(32, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(33, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(34, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(36, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(37, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(38, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(39, 2) = Worksheets("SD").Cells(6, 11)
```

```
Worksheets("Engine").Cells(40, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(41, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(42, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(43, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(44, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(45, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(46, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(47, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(48, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(49, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(50, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(51, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(52, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(53, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(54, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(55, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(56, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(57, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(58, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(59, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(60, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(61, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(62, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(63, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(64, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(65, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(66, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(67, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(68, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(69, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(70, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(71, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(72, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(73, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(74, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(75, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(76, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(77, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(78, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(79, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(80, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(81, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(82, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(83, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(84, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(85, 2) = Worksheets("SD").Cells(6, 11)
```

```
Worksheets("Engine").Cells(86, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(87, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(88, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(89, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(90, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(91, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(92, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(93, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(94, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(95, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(96, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(97, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(98, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(99, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(100, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(101, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(102, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(103, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(104, 2) = Worksheets("SD").Cells(6, 11)
Worksheets("Engine").Cells(110, 1) = Worksheets("SD").Cells(4, 11)
End Sub
```

This routine copies information from the solved worksheet engine to the worksheet table.

```
Sub Table2()
Range("A3:D104").Select
Selection.Copy
Sheets("Solved Table").Select
Range("H1").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
False, Transpose:=False
Range("A1").Select
End Sub
```

This routine copies information from the solved worksheet engine to the worksheet table.

```
Sub Table3()
Sheets("Engine").Select
Range("A5:A104").Select
Selection.Copy
Sheets("Solved Table").Select
Range("H3").Select
Selection.PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:= _
False, Transpose:=False
Sheets("Engine").Select
Range("A2").Select
End Sub
```

Appendix A: Transportation Problem

This routine updates the route worksheet based on the changes made on the input worksheet.

```
Sub Net()
miles
SandP
Application.DisplayAlerts = False
Worksheets("Routes").Delete
Set NewSheet = Worksheets.Add
NewSheet.Name = "Routes"
Worksheets("Routes").Move _
after:=Worksheets("Input")
back
MakeCircles
legend
Range("A2").Select
End Sub
```

This routine uses the Microsoft Solver Add-in to minimize the inputted values from the engine worksheet and activates the table subroutines. Before exiting this routine, the active worksheet is table.

```
Sub Solve()
Range("A2").Select
SandP
miles

Worksheets("Engine").Select

SolverSolve UserFinish:=False

Dim mySheet As Worksheet
Dim myshape As Shape
Set mySheet = Worksheets("Routes")
Dim Data30() As Variant
ReDim Data30(250)
Dim Data20() As Variant
ReDim Data20(250)
Dim Data21() As Variant
ReDim Data21(250)
Dim Data22() As Variant
ReDim Data22(250)
Dim Data23() As Variant
ReDim Data23(250)
Dim Data9() As Variant
ReDim Data9(250)
```

Appendix A: Transportation Problem

```
Dim Data10() As Variant
ReDim Data10(250)
Dim t
x = 5
y = 1
While Cells(x, y) <> ""
Data30(x) = Worksheets("Engine").Cells(x, y)
'MsgBox "Carry values are" & Data30(x)
x = x + 1   ' Increment Counter.
Wend

x = 5
y = 7
t = 0
While Worksheets("Engine").Cells(x, y) <> ""
Data10(x) = Worksheets("Engine").Cells(x, y)
If Data10(x) >= t Then
t = Data10(x)
'MsgBox "Total Destination nodes: " & T / 10
End If
'MsgBox "Destination node: " & Data9(x)
x = x + 1   ' Increment Counter.
Wend


x = 1
y = 67
While Worksheets("Engine").Cells(x, y) <> ""
Data20(x) = Worksheets("Engine").Cells(x, y)
'MsgBox "Destination node: " & Data20(x)
x = x + 1   ' Increment Counter.
Wend

x = 1
y = 68
While Worksheets("Engine").Cells(x, y) <> ""
Data21(x) = Worksheets("Engine").Cells(x, y)
x = x + 1   ' Increment Counter.
Wend
x = 1
y = 69
While Worksheets("Engine").Cells(x, y) <> ""
Data22(x) = Worksheets("Engine").Cells(x, y)
x = x + 1   ' Increment Counter.
Wend
x = 1
```

```
y = 70
While Worksheets("Engine").Cells(x, y) <> ""
Data23(x) = Worksheets("Engine").Cells(x, y)
x = x + 1   ' Increment Counter.

Wend

x = 5
y = 5
While Cells(x, y) <> ""
Data9(x) = Worksheets("Engine").Cells(x, y)

x = x + 1   ' Increment Counter.
Wend

For t = 5 To (x - 1)
Tempx = Data9(t)
For I = 1 To 200
If Data20(I) = Tempx Then
If Tempx <> 9.5 And Data30(t) >= 0.2 Then
Xcor = Data21(I)
End If
If Tempx = 9.5 And Data30(t) >= 0.2 Then
Xcor = Data21(I)
End If
End If
Next I

Tempy = Data10(t) / 10
For I = 1 To 200
If Data22(I) = Tempy And Data30(t) >= 0.2 Then
Ycor = Data23(I)
End If
Next I


With mySheet.Shapes.AddLine(25, Xcor, 320, Ycor).Line
.DashStyle = msoLineSolid
.ForeColor.RGB = RGB(225, 0, 0)
 End With
Next t
Table2
Table
Table3
Worksheets("Solved Table").Select
Range("A2").Select
```

```
End Sub
```

This routine adds the blue background to the route worksheet page.

```
Sub back()
Range("A1:G980").Select
ActiveWindow.ScrollColumn = 1
Range("H959").Select
ActiveWindow.ScrollRow = 1
Range("A1:AD550").Select
With Selection.Interior
.ColorIndex = 11
.Pattern = xlSolid
.PatternColorIndex = xlAutomatic
End With
Range("E10").Select
End Sub
```

Appendix B: Fuel Service Problem

```
Sub InitWrapper()

Sheet7.matrix2
Arrival
Service
Sim

If AddInIsPresent() Then
Application.Run "Q_Net.XLA!QNet_Init"
End If
End Sub
```

This routine initializes the output page.

```
Sub RunWrapper()

For I = 1 To 25
Worksheets("Output").Cells(5, 1) = 0
Worksheets("Output").Cells(5, 2) = 0
Worksheets("Output").Cells(5, 3) = 0
Worksheets("Output").Cells(5, 4) = 0
Worksheets("Output").Cells(5, 5) = 0
Worksheets("Output").Cells(5, 6) = 0
Worksheets("Output").Cells(5, 7) = 0
Next I
```

This routine runs the simulation macro 25 times, recording the information to the output worksheet.

```
For I = 1 To 25
Arrival
Service
Sim
Application.Run "Q_Net.XLA!QNet_Init"
Application.Run "Q_Net.XLA!RunButton_Pressed"

Worksheets("Output").Cells(5, 1) = Worksheets("Output").Cells(5, 1) +
Worksheets("Current").Cells(8, 9)
Worksheets("Output").Cells(5, 2) = Worksheets("Output").Cells(5, 2) + Worksheets("Current").Cells(8,
10)
Worksheets("Output").Cells(5, 3) = Worksheets("Output").Cells(5, 3) +
Worksheets("Current").Cells(8, 1)
Worksheets("Output").Cells(5, 4) = Worksheets("Output").Cells(5, 4) +
Worksheets("Current").Cells(8, 2)
Worksheets("Output").Cells(5, 5) = Worksheets("Output").Cells(5, 5) +
Worksheets("Current").Cells(8, 3)
```

Appendix B: Fuel Service Problem

```
Worksheets("Output").Cells(5, 6) = Worksheets("Output").Cells(5, 6) +
Worksheets("Current").Cells(8, 4)
Worksheets("Output").Cells(5, 7) = Worksheets("Output").Cells(5, 7) +
Worksheets("Current").Cells(8, 5)

If Worksheets("Output").Cells(5, 1) <> 0 Then
pos = 9
End If

If Worksheets("Output").Cells(5, 2) <> 0 Then
pos = 10
End If

If Worksheets("Output").Cells(5, 3) <> 0 Then
pos = 11
End If

If Worksheets("Output").Cells(5, 4) <> 0 Then
pos = 12
End If


If Worksheets("Output").Cells(5, 5) <> 0 Then
pos = 13
End If

If Worksheets("Output").Cells(5, 6) <> 0 Then
pos = 14
End If

If Worksheets("Output").Cells(5, 7) <> 0 Then
pos = 15
End If

Worksheets("Output").Cells(8 + I, 5) = Worksheets("Current").Cells(8, pos)
Next I
End Sub

Private Sub ToggleButton1_Click()
ActiveSheet.Shapes("Group 28").Select
Dim station As ShapeRange
Dim nCycle As Integer
If ToggleButton1.Value = True Then

Set myshape = ActiveSheet.Shapes("Group 28")
myshape.Fill.ForeColor.SchemeColor = 11
```

Appendix B: Fuel Service Problem

```
Set station = Sheet7.Shapes.Range(Array("Group 28"))
For nCycle = 1 To 100
station.IncrementTop -10
Next nCycle
station.IncrementRotation 10
For j = 1 To 100
Next j
station.IncrementRotation -10
Range("B14").Select
Else
Set station = Sheet7.Shapes.Range(Array("Group 28"))
For nCycle = 1 To 100
station.IncrementTop 10
Next nCycle
Range("B14").Select
ActiveSheet.Shapes("Group 28").Select
Set myshape = ActiveSheet.Shapes("Group 28")
myshape.Fill.ForeColor.SchemeColor = 23
End If
End Sub



Private Sub ToggleButton11_Click()
ActiveSheet.Shapes("Group 278").Select
Dim station As ShapeRange
Dim nCycle As Integer
If ToggleButton11.Value = True Then

Set myshape = ActiveSheet.Shapes("Group 278")
myshape.Fill.ForeColor.SchemeColor = 11

Set station = Sheet7.Shapes.Range(Array("Group 278"))
For nCycle = 1 To 70
station.IncrementTop -20


Next nCycle
station.IncrementRotation 10
DoEvents
For j = 1 To 100
Next j

station.IncrementRotation -10
Range("B14").Select
```

Appendix B: Fuel Service Problem

```
Else
Set station = Sheet7.Shapes.Range(Array("Group 278"))
For nCycle = 1 To 70
station.IncrementTop 20

Next nCycle
Set myshape = ActiveSheet.Shapes("Group 278")
myshape.Fill.ForeColor.SchemeColor = 23

Range("B14").Select
End If
End Sub

Private Sub ToggleButton12_Click()
ActiveSheet.Shapes("Group 678").Select
Dim station As ShapeRange
Dim nCycle As Integer

If ToggleButton12.Value = True Then

Set myshape = ActiveSheet.Shapes("Group 678")
myshape.Fill.ForeColor.SchemeColor = 11
Set station = Sheet7.Shapes.Range(Array("Group 678"))
For nCycle = 1 To 22
station.IncrementTop 2
station.IncrementRotation 180
If nCycle = 12 Or nCycle = 20 Then DoEvents
station.IncrementRotation -180

Next nCycle

Range("B14").Select
Else
Set station = Sheet7.Shapes.Range(Array("Group 678"))
For nCycle = 1 To 22
station.IncrementTop -2
Next nCycle
Set myshape = ActiveSheet.Shapes("Group 678")
myshape.Fill.ForeColor.SchemeColor = 8

Range("B14").Select
End If
End Sub

Private Sub ToggleButton2_Click()
ActiveSheet.Shapes("Group 428").Select
```

Appendix B: Fuel Service Problem

```
Dim station As ShapeRange
Dim nCycle As Integer

If ToggleButton2.Value = True Then

Set myshape = ActiveSheet.Shapes("Group 428")
myshape.Fill.ForeColor.SchemeColor = 11

Set station = Sheet7.Shapes.Range(Array("Group 428"))

For nCycle = 1 To 25
If nCycle = 22 Then DoEvents
station.IncrementLeft 2
Next nCycle

Range("B14").Select
Else
Set station = Sheet7.Shapes.Range(Array("Group 428"))
For nCycle = 1 To 25
station.IncrementLeft -2
Next nCycle
Set myshape = ActiveSheet.Shapes("Group 428")
myshape.Fill.ForeColor.SchemeColor = 8

Range("B14").Select
End If
End Sub


Private Sub ToggleButton3_Click()
ActiveSheet.Shapes("Group 528").Select
Dim station As ShapeRange
Dim nCycle As Integer

If ToggleButton3.Value = True Then

Set myshape = ActiveSheet.Shapes("Group 528")
myshape.Fill.ForeColor.SchemeColor = 11

Set station = Sheet7.Shapes.Range(Array("Group 528"))

For nCycle = 1 To 25
If nCycle = 22 Then DoEvents
station.IncrementLeft 2
Next nCycle
```

```
Range("B14").Select
Else
Set station = Sheet7.Shapes.Range(Array("Group 528"))
For nCycle = 1 To 25
station.IncrementLeft -2
Next nCycle
Set myshape = ActiveSheet.Shapes("Group 528")
myshape.Fill.ForeColor.SchemeColor = 8

Range("B14").Select
End If
End Sub

Private Sub ToggleButton5_Click()
ActiveSheet.Shapes("Group 1078").Select
Dim station As ShapeRange
Dim nCycle As Integer

If ToggleButton5.Value = True Then

Set myshape = ActiveSheet.Shapes("Group 1078")
myshape.Fill.ForeColor.SchemeColor = 11
Set station = Sheet7.Shapes.Range(Array("Group 1078"))
For nCycle = 1 To 22
station.IncrementTop 2
station.IncrementRotation 27
If nCycle = 12 Or nCycle = 20 Then DoEvents
station.IncrementRotation -27

Next nCycle

Range("G4").Select
Else
Set station = Sheet7.Shapes.Range(Array("Group 1078"))
For nCycle = 1 To 22
station.IncrementTop -2
Next nCycle
Set myshape = ActiveSheet.Shapes("Group 1078")
myshape.Fill.ForeColor.SchemeColor = 8

Range("G4").Select
End If
End Sub

Private Sub ToggleButton6_Click()
ActiveSheet.Shapes("Group 878").Select
```

Appendix B: Fuel Service Problem

```
Dim station As ShapeRange
Dim nCycle As Integer

If ToggleButton6.Value = True Then
matrix2
Set myshape = ActiveSheet.Shapes("Group 878")
myshape.Fill.ForeColor.SchemeColor = 11
Set station = Sheet7.Shapes.Range(Array("Group 878"))
For nCycle = 1 To 22
station.IncrementTop 2
station.IncrementRotation 270
If nCycle = 12 Or nCycle = 20 Then DoEvents
station.IncrementRotation -270

Next nCycle

Range("B14").Select
Else
Set station = Sheet7.Shapes.Range(Array("Group 878"))
For nCycle = 1 To 22
station.IncrementTop -2
Next nCycle
Set myshape = ActiveSheet.Shapes("Group 878")
myshape.Fill.ForeColor.SchemeColor = 8

Range("B14").Select
End If
End Sub
```

This routine initializes the priority queue in the transition matrix.

```
Sub matrix2()

Worksheets("Transition matrix").Cells(10, 15) = 0

Worksheets("Transition matrix").Cells(10, 14) = 0

Worksheets("Transition matrix").Cells(10, 13) = 0

Worksheets("Transition matrix").Cells(10, 12) = 0

Worksheets("Transition matrix").Cells(10, 11) = 0

Worksheets("Transition matrix").Cells(10, 10) = 0

Worksheets("Transition matrix").Cells(10, 9) = 0
```

Appendix B: Fuel Service Problem

```
If ToggleButton1.Value And ToggleButton2.Value And ToggleButton3.Value And
ToggleButton5.Value And ToggleButton6.Value And ToggleButton11.Value And
ToggleButton12.Value = "True" Then

Worksheets("Transition matrix").Cells(10, 15) =
"=IF(AND(Q_7<=Q_1,Q_7<=Q_2,Q_7<=Q_3,Q_7<=Q_4,Q_7<=Q_5,Q_7<=Q_6),1,0)"

Worksheets("Transition matrix").Cells(10, 14) =
"=IF(AND(Q_6<=Q_1,Q_6<=Q_2,Q_6<=Q_3,Q_6<=Q_4,Q_6<=Q_5,Q_6<Q_7),1,0)"

Worksheets("Transition matrix").Cells(10, 13) =
"=IF(AND(Q_5<=Q_1,Q_5<=Q_2,Q_5<=Q_3,Q_5<=Q_4,Q_5<Q_6,Q_5<Q_7),1,0)"

Worksheets("Transition matrix").Cells(10, 12) =
"=IF(AND(Q_4<=Q_1,Q_4<=Q_2,Q_4<=Q_3,Q_4<Q_5,Q_4<Q_6,Q_4<Q_7),1,0)"

Worksheets("Transition matrix").Cells(10, 11) =
"=IF(AND(Q_3<=Q_1,Q_3<=Q_2,Q_3<Q_4,Q_3<Q_5,Q_3<Q_6,Q_3<Q_7),1,0)"

Worksheets("Transition matrix").Cells(10, 10) =
"=IF(AND(Q_2<=Q_1,Q_2<Q_3,Q_2<Q_4,Q_2<Q_5,Q_2<Q_6,Q_2<Q_7),1,0)"

Worksheets("Transition matrix").Cells(10, 9) =
"=IF(AND(Q_1<Q_2,Q_1<Q_3,Q_1<Q_4,Q_1<Q_5,Q_1<Q_6,Q_1<Q_7),1,0)"
End If

If ToggleButton1.Value And ToggleButton2.Value And ToggleButton5.Value And
ToggleButton6.Value And ToggleButton11.Value And ToggleButton12.Value = "True"
And ToggleButton3.Value = "False" Then

Worksheets("Transition matrix").Cells(10, 14) =
"=IF(AND(Q_6<=Q_1,Q_6<=Q_2,Q_6<=Q_3,Q_6<=Q_4,Q_6<=Q_5),1,0)"

Worksheets("Transition matrix").Cells(10, 13) =
"=IF(AND(Q_5<=Q_1,Q_5<=Q_2,Q_5<=Q_3,Q_5<=Q_4,Q_5<Q_6),1,0)"

Worksheets("Transition matrix").Cells(10, 12) =
"=IF(AND(Q_4<=Q_1,Q_4<=Q_2,Q_4<=Q_3,Q_4<Q_5,Q_4<Q_6),1,0)"

Worksheets("Transition matrix").Cells(10, 11) =
"=IF(AND(Q_3<=Q_1,Q_3<=Q_2,Q_3<Q_4,Q_3<Q_5,Q_3<Q_6),1,0)"

Worksheets("Transition matrix").Cells(10, 10) =
"=IF(AND(Q_2<=Q_1,Q_2<Q_3,Q_2<Q_4,Q_2<Q_5,Q_2<Q_6),1,0)"
```

Appendix B: Fuel Service Problem

```
Worksheets("Transition matrix").Cells(10, 9) =
"=IF(AND(Q_1<Q_2,Q_1<Q_3,Q_1<Q_4,Q_1<Q_5,Q_1<Q_6),1,0)"

End If

If ToggleButton1.Value And ToggleButton2.Value And ToggleButton5.Value And
ToggleButton6.Value And ToggleButton12.Value = "True" And ToggleButton11.Value
= "False" Then

Worksheets("Transition matrix").Cells(10, 13) =
"=IF(AND(Q_5<=Q_1,Q_5<=Q_2,Q_5<=Q_3,Q_5<=Q_4),1,0)"

Worksheets("Transition matrix").Cells(10, 12) =
"=IF(AND(Q_4<=Q_1,Q_4<=Q_2,Q_4<=Q_3,Q_4<Q_5),1,0)"

Worksheets("Transition matrix").Cells(10, 11) =
"=IF(AND(Q_3<=Q_1,Q_3<=Q_2,Q_3<Q_4,Q_3<Q_5),1,0)"

Worksheets("Transition matrix").Cells(10, 10) =
"=IF(AND(Q_2<=Q_1,Q_2<Q_3,Q_2<Q_4,Q_2<Q_5),1,0)"

Worksheets("Transition matrix").Cells(10, 9) =
"=IF(AND(Q_1<Q_2,Q_1<Q_3,Q_1<Q_4,Q_1<Q_5),1,0)"

End If


If ToggleButton1.Value And ToggleButton2.Value And ToggleButton6.Value And
ToggleButton12.Value = "True" And ToggleButton5.Value = "False" Then


Worksheets("Transition matrix").Cells(10, 12) =
"=IF(AND(Q_4<=Q_1,Q_4<=Q_2,Q_4<=Q_3),1,0)"

Worksheets("Transition matrix").Cells(10, 11) =
"=IF(AND(Q_3<=Q_1,Q_3<=Q_2,Q_3<Q_4),1,0)"

Worksheets("Transition matrix").Cells(10, 10) =
"=IF(AND(Q_2<=Q_1,Q_2<Q_3,Q_2<Q_4),1,0)"

Worksheets("Transition matrix").Cells(10, 9) =
"=IF(AND(Q_1<Q_2,Q_1<Q_3,Q_1<Q_4),1,0)"

End If
```

Appendix B: Fuel Service Problem

```
If ToggleButton1.Value And ToggleButton2.Value And ToggleButton12.Value = "True"
And ToggleButton6.Value = "False" Then


Worksheets("Transition matrix").Cells(10, 11) =
"=IF(AND(Q_3<=Q_1,Q_3<=Q_2),1,0)"

Worksheets("Transition matrix").Cells(10, 10) = "=IF(AND(Q_2<=Q_1,Q_2<Q_3),1,0)"

Worksheets("Transition matrix").Cells(10, 9) = "=IF(AND(Q_1<Q_2,Q_1<Q_3),1,0)"

End If

If ToggleButton1.Value And ToggleButton2.Value = "True" And ToggleButton12.Value
= "False" Then


Worksheets("Transition matrix").Cells(10, 10) = "=IF(AND(Q_2<=Q_1),1,0)"

Worksheets("Transition matrix").Cells(10, 9) = "=IF(AND(Q_1<Q_2),1,0)"

End If

If ToggleButton1.Value = "True" And ToggleButton2.Value = "False" Then

Worksheets("Transition matrix").Cells(10, 9) = 1

End If

End Sub
```

This routine initializes the simulation time.

```
Sub Sim()
Worksheets("Current").Cells(7, 2) = Worksheets("Parallel").Cells(24, 7)
End Sub
```

This routine initializes the arrival time.

```
Sub Arrival()
Worksheets("Transition matrix").Cells(22, 8) = 1 / Worksheets("Parallel").Cells(21, 7)
End Sub
```

This routine initializes the service time for each queue.

Appendix B: Fuel Service Problem

```
Sub Service()
Worksheets("Transition matrix").Cells(22, 9) = Worksheets("Parallel").Cells(18, 4)
Worksheets("Transition matrix").Cells(22, 10) = Worksheets("Parallel").Cells(18, 5)
Worksheets("Transition matrix").Cells(22, 11) = Worksheets("Parallel").Cells(18, 6)
Worksheets("Transition matrix").Cells(22, 12) = Worksheets("Parallel").Cells(18, 7)
Worksheets("Transition matrix").Cells(22, 13) = Worksheets("Parallel").Cells(18, 8)
Worksheets("Transition matrix").Cells(22, 14) = Worksheets("Parallel").Cells(18, 9)
Worksheets("Transition matrix").Cells(22, 15) = Worksheets("Parallel").Cells(18, 10)
End Sub
```

Appendix C: Dining Facility Problem

This routine is for the arrival rate.

```
Sub Arrival()
Worksheets("Transition matrix").Cells(22, 8) = 1 / Worksheets("Parallel").Cells(21, 7)
End Sub
```

This routine sets the simulation time.

```
Sub Sim()
Worksheets("Current").Cells(7, 2) = Worksheets("Parallel").Cells(24, 7)
End Sub
```

This routine sets the service time.

```
Sub Service()
Worksheets("Transition matrix").Cells(22, 9) = Worksheets("Parallel").Cells(18, 4)
Worksheets("Transition matrix").Cells(22, 10) = Worksheets("Parallel").Cells(18, 5)
Worksheets("Transition matrix").Cells(22, 11) = Worksheets("Parallel").Cells(18, 6)
Worksheets("Transition matrix").Cells(22, 12) = Worksheets("Parallel").Cells(18, 7)
Worksheets("Transition matrix").Cells(22, 13) = Worksheets("Parallel").Cells(18, 8)
Worksheets("Transition matrix").Cells(22, 14) = Worksheets("Parallel").Cells(18, 9)
Worksheets("Transition matrix").Cells(22, 15) = Worksheets("Parallel").Cells(18, 10)
End Sub
```

This routine initializes the worksheet.

```
Sub InitWrapper()
Sheet7.matrix2
Arrival
Service
Sim
If AddInIsPresent() Then
Application.Run "Q_Net.XLA!QNet_Init"
End If
End Sub
```

This routine resets the worksheet for future replications.

```
Sub matrix2()
Worksheets("Transition matrix").Cells(16, 15) = 0
Worksheets("Transition matrix").Cells(15, 14) = 0
Worksheets("Transition matrix").Cells(14, 13) = 0
Worksheets("Transition matrix").Cells(13, 12) = 0
Worksheets("Transition matrix").Cells(12, 11) = 0
Worksheets("Transition matrix").Cells(11, 10) = 0
Worksheets("Transition matrix").Cells(10, 9) = 0
```

Appendix C: Dining Facility Problem

These "IF" statements checks to see which macro buttons are active on the worksheet to set the appropriate values on the transition matrix.

```
If ToggleButton1.value And ToggleButton2.value And ToggleButton3.value And
ToggleButton5.value And ToggleButton6.value And ToggleButton11.value And
ToggleButton12.value = "True" Then

Worksheets("Transition matrix").Cells(16, 15) = "= 1"
Worksheets("Transition matrix").Cells(15, 14) = "= 1"
Worksheets("Transition matrix").Cells(14, 13) = "= 1"
Worksheets("Transition matrix").Cells(13, 12) = "= 1"
Worksheets("Transition matrix").Cells(12, 11) = "= 1"
Worksheets("Transition matrix").Cells(11, 10) = "= 1"
Worksheets("Transition matrix").Cells(10, 9) = "= 1"
End If

If ToggleButton1.value And ToggleButton2.value And ToggleButton5.value And
ToggleButton6.value And ToggleButton11.value And ToggleButton12.value = "True"
And ToggleButton3.value = "False" Then

Worksheets("Transition matrix").Cells(15, 14) = "= 1"
Worksheets("Transition matrix").Cells(14, 13) = "= 1"
Worksheets("Transition matrix").Cells(13, 12) = "= 1"
Worksheets("Transition matrix").Cells(12, 11) = "= 1"
Worksheets("Transition matrix").Cells(11, 10) = "= 1"
Worksheets("Transition matrix").Cells(10, 9) = "= 1"

End If

If ToggleButton1.value And ToggleButton2.value And ToggleButton5.value And
ToggleButton6.value And ToggleButton12.value = "True" And ToggleButton11.value =
"False" Then

Worksheets("Transition matrix").Cells(14, 13) = "=1"
Worksheets("Transition matrix").Cells(13, 12) = "= 1"
Worksheets("Transition matrix").Cells(12, 11) = "= 1"
Worksheets("Transition matrix").Cells(11, 10) = "= 1"
Worksheets("Transition matrix").Cells(10, 9) = "= 1"

End If


If ToggleButton1.value And ToggleButton2.value And ToggleButton6.value And
ToggleButton12.value = "True" And ToggleButton5.value = "False" Then
```

```
Worksheets("Transition matrix").Cells(13, 12) = "= 1"
Worksheets("Transition matrix").Cells(12, 11) = "= 1"
Worksheets("Transition matrix").Cells(11, 10) = "= 1"
Worksheets("Transition matrix").Cells(10, 9) = "= 1"

End If

If ToggleButton1.value And ToggleButton2.value And ToggleButton12.value = "True"
And ToggleButton6.value = "False" Then


Worksheets("Transition matrix").Cells(12, 11) = "= 1"
Worksheets("Transition matrix").Cells(11, 10) = "= 1"
Worksheets("Transition matrix").Cells(10, 9) = "= 1"
End If

If (ToggleButton1.value And ToggleButton2.value = True) And ToggleButton12.value =
"False" Then

Worksheets("Transition matrix").Cells(11, 10) = "= 1"
Worksheets("Transition matrix").Cells(10, 9) = "= 1"

End If

If ToggleButton1.value = True And ToggleButton2.value = "False" Then
Worksheets("Transition matrix").Cells(10, 9) = 1
End If

End Sub
```

This routine runs the simulation a total of 25 replications, and it records the information to an output file for analysis.

```
Sub RunWrapper()
'MsgBox "run is equal to" & Worksheets("Parallel").Cells(41, 8)
'For I = 1 To Worksheets("Parallel").Cells(41, 8)
For I = 1 To 25
Worksheets("Output").Cells(5, 1) = 0
Worksheets("Output").Cells(5, 2) = 0
Worksheets("Output").Cells(5, 3) = 0
Worksheets("Output").Cells(5, 4) = 0
Worksheets("Output").Cells(5, 5) = 0h
Worksheets("Output").Cells(5, 6) = 0
Worksheets("Output").Cells(5, 7) = 0
Number = 0
Next I
```

Appendix C: Dining Facility Problem

```
For I = 1 To 25
Arrival
Service
Sim
Number = 0
Application.Run "Q_Net.XLA!QNet_Init"
Application.Run "Q_Net.XLA!RunButton_Pressed"
Worksheets("Output").Cells(5, 1) = Worksheets("Output").Cells(5, 1) +
Worksheets("Current").Cells(8, 9)
Worksheets("Output").Cells(5, 2) = Worksheets("Output").Cells(5, 2) +
Worksheets("Current").Cells(8, 10)
Worksheets("Output").Cells(5, 3) = Worksheets("Output").Cells(5, 3) +
Worksheets("Current").Cells(8, 11)
Worksheets("Output").Cells(5, 4) = Worksheets("Output").Cells(5, 4) +
Worksheets("Current").Cells(8, 12)
Worksheets("Output").Cells(5, 5) = Worksheets("Output").Cells(5, 5) +
Worksheets("Current").Cells(8, 13)
Worksheets("Output").Cells(5, 6) = Worksheets("Output").Cells(5, 6) +
Worksheets("Current").Cells(8, 14)
Worksheets("Output").Cells(5, 7) = Worksheets("Output").Cells(5, 7) +
Worksheets("Current").Cells(8, 15)
If Worksheets("Output").Cells(5, 1) <> 0 Then
pos = 9
Number = Number + 1
End If
If Worksheets("Output").Cells(5, 2) <> 0 Then
pos = 10
Number = Number + 1
End If

If Worksheets("Output").Cells(5, 3) <> 0 Then
pos = 11
Number = Number + 1
End If

If Worksheets("Output").Cells(5, 4) <> 0 Then
pos = 12
Number = Number + 1
End If

If Worksheets("Output").Cells(5, 5) <> 0 Then
pos = 13
Number = Number + 1
End If
If Worksheets("Output").Cells(5, 6) <> 0 Then
```

Appendix C: Dining Facility Problem

```
pos = 14
Number = Number + 1
End If

If Worksheets("Output").Cells(5, 7) <> 0 Then
pos = 15
Number = Number + 1
End If
Worksheets("Output").Cells(8 + I, 6) = (Worksheets("Current").Cells(8, 9) +
Worksheets("Current").Cells(8, 10) + Worksheets("Current").Cells(8, 11) +
Worksheets("Current").Cells(8, 12) + Worksheets("Current").Cells(8, 13) +
Worksheets("Current").Cells(8, 14) + Worksheets("Current").Cells(8, 15))
Worksheets("Output").Cells(8 + I, 5) = Worksheets("Current").Cells(6, pos)
Next I
MsgBox "number is " & Number
MsgBox "pos is" & pos
value
sort
End Sub
```

Appendix D: Risk Management Worksheet

The private routines initialize the combo boxes on the risk management worksheet.

```
Private Sub ComboBox1_Change()
End Sub

Private Sub ComboBox10_Change()
End Sub

Private Sub ComboBox100_Change()
End Sub

Private Sub ComboBox101_Change()
End Sub

Private Sub ComboBox102_Change()
End Sub

Private Sub ComboBox103_Change()
End Sub

Private Sub ComboBox104_Change()
End Sub

Private Sub ComboBox105_Change()
End Sub

Private Sub ComboBox106_Change()
End Sub

Private Sub ComboBox107_Change()
End Sub

Private Sub ComboBox108_Change()
End Sub

Private Sub ComboBox109_Change()
End Sub

Private Sub ComboBox11_Change()
End Sub

Private Sub ComboBox110_Change()

End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub ComboBox111_Change()
End Sub

Private Sub ComboBox112_Change()
End Sub

Private Sub ComboBox113_Change()
End Sub

Private Sub ComboBox114_Change()
End Sub

Private Sub ComboBox115_Change()
End Sub

Private Sub ComboBox116_Change()
End Sub

Private Sub ComboBox117_Change()
End Sub

Private Sub ComboBox118_Change()
End Sub

Private Sub ComboBox119_Change()
End Sub

Private Sub ComboBox12_Change()
End Sub

Private Sub ComboBox120_Change()
End Sub

Private Sub ComboBox13_Change()
End Sub

Private Sub ComboBox14_Change()
End Sub

Private Sub ComboBox15_Change()
End Sub

Private Sub ComboBox16_Change()
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub ComboBox17_Change()
End Sub

Private Sub ComboBox18_Change()
End Sub

Private Sub ComboBox19_Change()
End Sub

Private Sub ComboBox2_Change()
End Sub

Private Sub ComboBox20_Change()
End Sub

Private Sub ComboBox21_Change()
End Sub

Private Sub ComboBox22_Change()
End Sub

Private Sub ComboBox23_Change()
End Sub

Private Sub ComboBox24_Change()
End Sub

Private Sub ComboBox25_Change()
End Sub

Private Sub ComboBox26_Change()
End Sub

Private Sub ComboBox27_Change()
End Sub

Private Sub ComboBox28_Change()
End Sub

Private Sub ComboBox29_Change()
End Sub

Private Sub ComboBox3_Change()
End Sub
```

```
Private Sub ComboBox30_Change()
End Sub

Private Sub ComboBox31_Change()
End Sub

Private Sub ComboBox32_Change()
End Sub

Private Sub ComboBox33_Change()
End Sub

Private Sub ComboBox34_Change()
End Sub

Private Sub ComboBox35_Change()
End Sub

Private Sub ComboBox36_Change()
End Sub

Private Sub ComboBox37_Change()
End Sub

Private Sub ComboBox38_Change()
End Sub

Private Sub ComboBox39_Change()
End Sub

Private Sub ComboBox4_Change()
End Sub

Private Sub ComboBox40_Change()
End Sub

Private Sub ComboBox41_Change()
End Sub

Private Sub ComboBox42_Change()
End Sub

Private Sub ComboBox43_Change()
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub ComboBox44_Change()
End Sub

Private Sub ComboBox45_Change()
End Sub

Private Sub ComboBox46_Change()
End Sub

Private Sub ComboBox47_Change()
End Sub

Private Sub ComboBox48_Change()
End Sub

Private Sub ComboBox49_Change()
End Sub

Private Sub ComboBox5_Change()
End Sub

Private Sub ComboBox50_Change()
End Sub

Private Sub ComboBox51_Change()
End Sub

Private Sub ComboBox52_Change()
End Sub

Private Sub ComboBox53_Change()
End Sub

Private Sub ComboBox54_Change()
End Sub

Private Sub ComboBox55_Change()
End Sub

Private Sub ComboBox56_Change()
End Sub

Private Sub ComboBox57_Change()
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub ComboBox58_Change()
End Sub

Private Sub ComboBox59_Change()
End Sub

Private Sub ComboBox6_Change()
End Sub

Private Sub ComboBox60_Change()
End Sub

Private Sub ComboBox61_Change()
End Sub


Private Sub ComboBox62_Change()
End Sub

Private Sub ComboBox63_Change()
End Sub

Private Sub ComboBox64_Change()
End Sub

Private Sub ComboBox65_Change()
End Sub

Private Sub ComboBox66_Change()
End Sub

Private Sub ComboBox67_Change()
End Sub

Private Sub ComboBox68_Change()
End Sub

Private Sub ComboBox69_Change()
End Sub


Private Sub ComboBox7_Change()
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub ComboBox71_Change()
End Sub

Private Sub ComboBox8_Change()
End Sub

Private Sub ComboBox80_Change()
End Sub

Private Sub ComboBox81_Change()
End Sub

Private Sub ComboBox82_Change()
End Sub

Private Sub ComboBox83_Change()
End Sub

Private Sub ComboBox84_Change()
End Sub

Private Sub ComboBox85_Change()
End Sub

Private Sub ComboBox86_Change()
End Sub

Private Sub ComboBox87_Change()
End Sub

Private Sub ComboBox88_Change()
End Sub

Private Sub ComboBox89_Change()
End Sub

Private Sub ComboBox9_Change()
End Sub

Private Sub ComboBox90_Change()
End Sub

Private Sub ComboBox91_Change()
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub ComboBox92_Change()
End Sub

Private Sub ComboBox93_Change()
End Sub

Private Sub ComboBox94_Change()
End Sub

Private Sub ComboBox95_Change()
End Sub

Private Sub ComboBox96_Change()
End Sub

Private Sub ComboBox97_Change()
End Sub

Private Sub ComboBox98_Change()
End Sub

Private Sub ComboBox99_Change()
End Sub
```

This "initialize" command macro button has two functional purposes. It clears any information in the combo boxes on the risk management worksheet by using the clear feature. Finally, it adds information to each combo box by the use of the add item feature.

```
Private Sub CommandButton1_Click()
Worksheets("Analysis Sheet").Select
ComboBox1.Clear
ComboBox2.Clear
ComboBox3.Clear
ComboBox4.Clear
ComboBox5.Clear
ComboBox6.Clear
ComboBox7.Clear
ComboBox8.Clear
ComboBox9.Clear
ComboBox10.Clear
ComboBox11.Clear
ComboBox12.Clear
ComboBox13.Clear
```

```
ComboBox14.Clear
ComboBox15.Clear
ComboBox16.Clear
ComboBox17.Clear
ComboBox18.Clear
ComboBox19.Clear
ComboBox20.Clear
ComboBox21.Clear
ComboBox22.Clear
ComboBox23.Clear
ComboBox24.Clear
ComboBox25.Clear
ComboBox26.Clear
ComboBox27.Clear
ComboBox28.Clear
ComboBox29.Clear
ComboBox30.Clear
ComboBox31.Clear
ComboBox32.Clear
ComboBox33.Clear
ComboBox34.Clear
ComboBox35.Clear
ComboBox36.Clear
ComboBox37.Clear
ComboBox38.Clear
ComboBox39.Clear
ComboBox40.Clear
For x = 1 To 1
ComboBox1.AddItem "L"
ComboBox1.AddItem "M"
ComboBox1.AddItem "H"
ComboBox1.AddItem "EH"

ComboBox2.AddItem "L"
ComboBox2.AddItem "M"
ComboBox2.AddItem "H"
ComboBox2.AddItem "EH"

ComboBox3.AddItem "L"
ComboBox3.AddItem "M"
ComboBox3.AddItem "H"
ComboBox3.AddItem "EH"

ComboBox4.AddItem "L"
ComboBox4.AddItem "M"
```

```
ComboBox4.AddItem "H"
ComboBox4.AddItem "EH"


ComboBox5.AddItem "L"
ComboBox5.AddItem "M"
ComboBox5.AddItem "H"
ComboBox5.AddItem "EH"

ComboBox6.AddItem "L"
ComboBox6.AddItem "M"
ComboBox6.AddItem "H"
ComboBox6.AddItem "EH"

ComboBox7.AddItem "L"
ComboBox7.AddItem "M"
ComboBox7.AddItem "H"
ComboBox7.AddItem "EH"

ComboBox8.AddItem "L"
ComboBox8.AddItem "M"
ComboBox8.AddItem "H"
ComboBox8.AddItem "EH"


ComboBox9.AddItem "L"
ComboBox9.AddItem "M"
ComboBox9.AddItem "H"
ComboBox9.AddItem "EH"

ComboBox10.AddItem "L"
ComboBox10.AddItem "M"
ComboBox10.AddItem "H"
ComboBox10.AddItem "EH"

ComboBox11.AddItem "L"
ComboBox11.AddItem "M"
ComboBox11.AddItem "H"
ComboBox11.AddItem "EH"

ComboBox12.AddItem "L"
ComboBox12.AddItem "M"
ComboBox12.AddItem "H"
ComboBox12.AddItem "EH"
```

Appendix D: Risk Management Worksheet

```
ComboBox13.AddItem "L"
ComboBox13.AddItem "M"
ComboBox13.AddItem "H"
ComboBox13.AddItem "EH"

ComboBox14.AddItem "L"
ComboBox14.AddItem "M"
ComboBox14.AddItem "H"
ComboBox14.AddItem "EH"

ComboBox15.AddItem "L"
ComboBox15.AddItem "M"
ComboBox15.AddItem "H"
ComboBox15.AddItem "EH"

ComboBox16.AddItem "L"
ComboBox16.AddItem "M"
ComboBox16.AddItem "H"
ComboBox16.AddItem "EH"

ComboBox17.AddItem "L"
ComboBox17.AddItem "M"
ComboBox17.AddItem "H"
ComboBox17.AddItem "EH"

ComboBox18.AddItem "L"
ComboBox18.AddItem "M"
ComboBox18.AddItem "H"
ComboBox18.AddItem "EH"

ComboBox19.AddItem "L"
ComboBox19.AddItem "M"
ComboBox19.AddItem "H"
ComboBox19.AddItem "EH"

ComboBox20.AddItem "L"
ComboBox20.AddItem "M"
ComboBox20.AddItem "H"
ComboBox20.AddItem "EH"

ComboBox21.AddItem "L"
ComboBox21.AddItem "M"
ComboBox21.AddItem "H"
ComboBox21.AddItem "EH"
```

```
ComboBox22.AddItem "L"
ComboBox22.AddItem "M"
ComboBox22.AddItem "H"
ComboBox22.AddItem "EH"

ComboBox23.AddItem "L"
ComboBox23.AddItem "M"
ComboBox23.AddItem "H"
ComboBox23.AddItem "EH"

ComboBox24.AddItem "L"
ComboBox24.AddItem "M"
ComboBox24.AddItem "H"
ComboBox24.AddItem "EH"

ComboBox25.AddItem "L"
ComboBox25.AddItem "M"
ComboBox25.AddItem "H"
ComboBox25.AddItem "EH"

ComboBox26.AddItem "L"
ComboBox26.AddItem "M"
ComboBox26.AddItem "H"
ComboBox26.AddItem "EH"

ComboBox27.AddItem "L"
ComboBox27.AddItem "M"
ComboBox27.AddItem "H"
ComboBox27.AddItem "EH"

ComboBox28.AddItem "L"
ComboBox28.AddItem "M"
ComboBox28.AddItem "H"
ComboBox28.AddItem "EH"

ComboBox29.AddItem "L"
ComboBox29.AddItem "M"
ComboBox29.AddItem "H"
ComboBox29.AddItem "EH"

ComboBox30.AddItem "L"
ComboBox30.AddItem "M"
ComboBox30.AddItem "H"
ComboBox30.AddItem "EH"
```

```
ComboBox31.AddItem "L"
ComboBox31.AddItem "M"
ComboBox31.AddItem "H"
ComboBox31.AddItem "EH"

ComboBox32.AddItem "L"
ComboBox32.AddItem "M"
ComboBox32.AddItem "H"
ComboBox32.AddItem "EH"

ComboBox33.AddItem "L"
ComboBox33.AddItem "M"
ComboBox33.AddItem "H"
ComboBox33.AddItem "EH"

ComboBox34.AddItem "L"
ComboBox34.AddItem "M"
ComboBox34.AddItem "H"
ComboBox34.AddItem "EH"

ComboBox35.AddItem "L"
ComboBox35.AddItem "M"
ComboBox35.AddItem "H"
ComboBox35.AddItem "EH"

ComboBox36.AddItem "L"
ComboBox36.AddItem "M"
ComboBox36.AddItem "H"
ComboBox36.AddItem "EH"

ComboBox37.AddItem "L"
ComboBox37.AddItem "M"
ComboBox37.AddItem "H"
ComboBox37.AddItem "EH"

ComboBox38.AddItem "L"
ComboBox38.AddItem "M"
ComboBox38.AddItem "H"
ComboBox38.AddItem "EH"

ComboBox39.AddItem "L"
ComboBox39.AddItem "M"
ComboBox39.AddItem "H"
ComboBox39.AddItem "EH"
```

```
ComboBox40.AddItem "L"
ComboBox40.AddItem "M"
ComboBox40.AddItem "H"
ComboBox40.AddItem "EH"
Next x

End Sub
```

This command button will calculate the average, mode, and highest occurrence of a risk factor based on the input from the risk management worksheet.

```
Private Sub CommandButton2_Click()
Probsum = 0
Count = 0
Low = 0
Med = 0
High = 0
Ehigh = 0

If ComboBox21.Value = "L" Then
Count = Count + 1
Low = Low + 1

Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox21.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox21.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox21.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox22.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)
```

```
ElseIf ComboBox22.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox22.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox22.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox23.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox23.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox23.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox23.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox24.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox24.Value = "M" Then
Count = Count + 1
Med = Med + 1
```

```
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox24.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox24.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox25.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox25.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox25.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox25.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox26.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox26.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox26.Value = "H" Then
```

Appendix D: Risk Management Worksheet

```
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox26.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox27.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox27.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox27.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox27.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If



If ComboBox28.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox28.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox28.Value = "H" Then
```

```
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox28.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox29.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox29.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox29.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox29.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If



If ComboBox30.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox30.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox30.Value = "H" Then
Count = Count + 1
```

```
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox30.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox31.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox31.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox31.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox31.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox32.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox32.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox32.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)
```

Appendix D: Risk Management Worksheet

```
ElseIf ComboBox32.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox33.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox33.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox33.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox33.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox34.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox34.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox34.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox34.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
```

```
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox35.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox35.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)


ElseIf ComboBox35.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox35.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox36.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox36.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox36.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox36.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If
```

```
If ComboBox37.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox37.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox37.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox37.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If


If ComboBox38.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox38.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox38.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox38.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If
```

Appendix D: Risk Management Worksheet

```
If ComboBox39.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox39.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox39.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox39.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox40.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox40.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox40.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox40.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

Mode = Low

If Med >= Mode Then
```

```
Mode = Med
End If

If High >= Mode Then
Mode = High
End If

If Ehigh >= Mode Then
Mode = Ehigh
End If


If Mode = Ehigh Then
MsgBox "Assessment 1: The overall mode is Extremly High."
ElseIf Mode = High Then
MsgBox "Assessment 1: The overall mode is High."
ElseIf Mode = Med Then
MsgBox "Assessment 1: The overall mode is Moderate."
ElseIf Mode = Low Then
MsgBox "Assessment 1: The overall mode is Low."
End If


If Count <> 0 Then
T1 = 0
T2 = 0
T3 = 0
T4 = 0
Value = Probsum / Count
T1 = Abs(Value - Worksheets("Initial data").Cells(3, 6))
T2 = Abs(Value - Worksheets("Initial data").Cells(4, 6))
T3 = Abs(Value - Worksheets("Initial data").Cells(5, 6))
T4 = Abs(Value - Worksheets("Initial data").Cells(6, 6))

If (T1 < T2 And T1 < T3 And T1 < T4) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is Low (L), and the average outcome for mission/task level before controls
are implemented is   ") & Value
End If

If (T2 <= T1 And T2 < T3 And T2 < T4) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is Moderate (M), and the average outcome for mission/task level before
controls are implemented is   ") & Value
End If
```

```
If (T3 <= T1 And T3 <= T2 And T3 < T4) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is High (H), and the average outcome for mission/task level before controls
are implemented is   ") & Value
End If

If (T4 <= T1 And T4 <= T2 And T4 <= T3) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is Extremely High (EH), and the average outcome for mission/task level
before controls are implemented is   ") & Value
End If

If (Ehigh > 0) Then MsgBox "Assessment 3: The overall risk is (EH) extremely high."
If (High > 0 And Ehigh = 0) Then MsgBox "Assessment 3: The overall risk is (H) High."
If (Med > 0 And Ehigh = 0 And High = 0) Then MsgBox "Assessment 3: The overall risk
is (M) Moderate."
If (Low > 0 And Ehigh = 0 And High = 0 And Med = 0) Then MsgBox "Assessment 3:
The overall risk is (L) Low."
End If
End Sub
```

This "clear " command macro button will clear the entire risk management worksheet.

```
Private Sub CommandButton3_Click()
TextBox1.Value = " Double click on space to enter text"
TextBox2.Value = " Double click on space to enter text"
TextBox3.Value = " Double click on space to enter text"
TextBox4.Value = " Double click on space to enter text"
TextBox5.Value = " Double click on space to enter text"
TextBox6.Value = " Double click on space to enter text"
TextBox7.Value = " Double click on space to enter text"
TextBox8.Value = " Double click on space to enter text"
TextBox9.Value = " Double click on space to enter text"
TextBox10.Value = " Double click on space to enter text"
TextBox11.Value = " Double click on space to enter text"
TextBox12.Value = " Double click on space to enter text"
TextBox13.Value = " Double click on space to enter text"
TextBox14.Value = " Double click on space to enter text"
TextBox15.Value = " Double click on space to enter text"
TextBox16.Value = " Double click on space to enter text"
TextBox17.Value = " Double click on space to enter text"
TextBox18.Value = " Double click on space to enter text"
```

Appendix D: Risk Management Worksheet

```
TextBox19.Value = " Double click on space to enter text"
TextBox20.Value = " Double click on space to enter text"

ComboBox1.Value = " "
ComboBox2.Value = " "
ComboBox3.Value = " "
ComboBox4.Value = " "
ComboBox5.Value = " "
ComboBox6.Value = " "
ComboBox7.Value = " "
ComboBox8.Value = " "
ComboBox9.Value = " "
ComboBox10.Value = " "
ComboBox11.Value = " "
ComboBox12.Value = " "
ComboBox13.Value = " "
ComboBox14.Value = " "
ComboBox15.Value = " "
ComboBox16.Value = " "
ComboBox17.Value = " "
ComboBox18.Value = " "
ComboBox19.Value = " "
ComboBox20.Value = " "

ComboBox21.Value = " "
ComboBox22.Value = " "
ComboBox23.Value = " "
ComboBox24.Value = " "
ComboBox25.Value = " "
ComboBox26.Value = " "
ComboBox27.Value = " "
ComboBox28.Value = " "
ComboBox29.Value = " "
ComboBox30.Value = " "
ComboBox31.Value = " "
ComboBox32.Value = " "
ComboBox33.Value = " "
ComboBox34.Value = " "
ComboBox35.Value = " "
ComboBox36.Value = " "
ComboBox37.Value = " "
ComboBox38.Value = " "
ComboBox39.Value = " "
ComboBox40.Value = " "
```

Appendix D: Risk Management Worksheet

```
ComboBox61.Value = " "
ComboBox62.Value = " "
ComboBox63.Value = " "
ComboBox64.Value = " "
ComboBox65.Value = " "
ComboBox66.Value = " "
ComboBox67.Value = " "
ComboBox68.Value = " "
ComboBox69.Value = " "
ComboBox70.Value = " "
ComboBox71.Value = " "
ComboBox72.Value = " "
ComboBox73.Value = " "
ComboBox74.Value = " "
ComboBox75.Value = " "
ComboBox76.Value = " "
ComboBox77.Value = " "
ComboBox78.Value = " "
ComboBox79.Value = " "
ComboBox80.Value = " "

ComboBox81.Value = " "
ComboBox82.Value = " "
ComboBox83.Value = " "
ComboBox84.Value = " "
ComboBox85.Value = " "
ComboBox86.Value = " "
ComboBox87.Value = " "
ComboBox88.Value = " "
ComboBox89.Value = " "
ComboBox90.Value = " "
ComboBox91.Value = " "
ComboBox92.Value = " "
ComboBox93.Value = " "
ComboBox94.Value = " "
ComboBox95.Value = " "
ComboBox96.Value = " "
ComboBox97.Value = " "
ComboBox98.Value = " "
ComboBox99.Value = " "
ComboBox100.Value = " "

ComboBox41.Value = " "
ComboBox42.Value = " "
ComboBox43.Value = " "
```

```
ComboBox44.Value = " "
ComboBox45.Value = " "
ComboBox46.Value = " "
ComboBox47.Value = " "
ComboBox48.Value = " "
ComboBox49.Value = " "
ComboBox50.Value = " "
ComboBox51.Value = " "
ComboBox52.Value = " "
ComboBox53.Value = " "
ComboBox54.Value = " "
ComboBox55.Value = " "
ComboBox56.Value = " "
ComboBox57.Value = " "
ComboBox58.Value = " "
ComboBox59.Value = " "
ComboBox60.Value = " "

ComboBox61.Clear
ComboBox62.Clear
ComboBox63.Clear
ComboBox64.Clear
ComboBox65.Clear
ComboBox66.Clear
ComboBox67.Clear
ComboBox68.Clear
ComboBox69.Clear
ComboBox70.Clear
ComboBox71.Clear
ComboBox72.Clear
ComboBox73.Clear
ComboBox74.Clear
ComboBox75.Clear
ComboBox76.Clear
ComboBox77.Clear
ComboBox78.Clear
ComboBox79.Clear
ComboBox80.Clear
ComboBox81.Clear
ComboBox82.Clear
ComboBox83.Clear
ComboBox84.Clear
ComboBox85.Clear
ComboBox86.Clear
ComboBox87.Clear
```

```
ComboBox88.Clear
ComboBox89.Clear
ComboBox90.Clear
ComboBox91.Clear
ComboBox92.Clear
ComboBox93.Clear
ComboBox94.Clear
ComboBox95.Clear
ComboBox96.Clear
ComboBox97.Clear
ComboBox98.Clear
ComboBox99.Clear
ComboBox100.Clear
ComboBox41.Clear
ComboBox42.Clear
ComboBox43.Clear
ComboBox44.Clear
ComboBox45.Clear
ComboBox46.Clear
ComboBox47.Clear
ComboBox48.Clear
ComboBox49.Clear
ComboBox50.Clear
ComboBox51.Clear
ComboBox52.Clear
ComboBox53.Clear
ComboBox54.Clear
ComboBox55.Clear
ComboBox56.Clear
ComboBox57.Clear
ComboBox58.Clear
ComboBox59.Clear
ComboBox60.Clear
End Sub
```

This command button will calculate the average, mode, and highest occurrence of a risk factor based on the input from the risk management worksheet.

```
Private Sub CommandButton4_Click()
Probsum = 0
Count = 0
Low = 0
Med = 0
High = 0
```

Appendix D: Risk Management Worksheet

```
Ehigh = 0

If ComboBox1.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox1.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox1.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox1.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox2.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox2.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox2.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox2.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If
```

Appendix D: Risk Management Worksheet

```
If ComboBox3.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox3.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox3.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox3.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox4.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox4.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox4.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox4.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox5.Value = "L" Then
Count = Count + 1
Low = Low + 1
```

```
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox5.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox5.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox5.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox6.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox6.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox6.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox6.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox7.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox7.Value = "M" Then
```

```
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox7.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox7.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If



If ComboBox8.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox8.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox8.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox8.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox9.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox9.Value = "M" Then
```

```
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox9.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox9.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox10.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox10.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox10.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox10.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox11.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox11.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)
```

```
ElseIf ComboBox11.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox11.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox12.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox12.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox12.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox12.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox13.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox13.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox13.Value = "H" Then
Count = Count + 1
```

```
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox13.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox14.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox14.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox14.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox14.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox15.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox15.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox15.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)
```

```
ElseIf ComboBox15.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox16.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox16.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox16.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox16.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)

End If

If ComboBox17.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox17.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox17.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox17.Value = "EH" Then
Count = Count + 1
```

```
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox18.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox18.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox18.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox18.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

If ComboBox19.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox19.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox19.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox19.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If
```

Appendix D: Risk Management Worksheet

```
If ComboBox20.Value = "L" Then
Count = Count + 1
Low = Low + 1
Probsum = Probsum + Worksheets("Initial data").Cells(3, 6)

ElseIf ComboBox20.Value = "M" Then
Count = Count + 1
Med = Med + 1
Probsum = Probsum + Worksheets("Initial data").Cells(4, 6)

ElseIf ComboBox20.Value = "H" Then
Count = Count + 1
High = High + 1
Probsum = Probsum + Worksheets("Initial data").Cells(5, 6)

ElseIf ComboBox20.Value = "EH" Then
Count = Count + 1
Ehigh = Ehigh + 1
Probsum = Probsum + Worksheets("Initial data").Cells(6, 6)
End If

Mode = Low

If Med >= Mode Then
Mode = Med
End If

If High >= Mode Then
Mode = High
End If

If Ehigh >= Mode Then
Mode = Ehigh
End If


If Mode = Ehigh Then
MsgBox "Assessment 1: The overall mode is Extremely High."

ElseIf Mode = High Then
MsgBox "Assessment 1: The overall mode is High."

ElseIf Mode = Med Then
MsgBox "Assessment 1: The overall mode is Moderate."
```

Appendix D: Risk Management Worksheet

```
ElseIf Mode = Low Then
MsgBox "Assessment 1: The overall mode is Low."
End If


If Count <> 0 Then
T1 = 0
T2 = 0
T3 = 0
T4 = 0
Value = Probsum / Count
T1 = Abs(Value - Worksheets("Initial data").Cells(3, 6))
T2 = Abs(Value - Worksheets("Initial data").Cells(4, 6))
T3 = Abs(Value - Worksheets("Initial data").Cells(5, 6))
T4 = Abs(Value - Worksheets("Initial data").Cells(6, 6))

If (T1 < T2 And T1 < T3 And T1 < T4) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is Low (L), and the average outcome for mission/task level before controls
are implemented is   ") & Value
End If

If (T2 <= T1 And T2 < T3 And T2 < T4) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is Moderate (M), and the average outcome for mission/task level before
controls are implemented is   ") & Value
End If

If (T3 <= T1 And T3 <= T2 And T3 < T4) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is High (H), and the average outcome for mission/task level before controls
are implemented is   ") & Value
End If

If (T4 <= T1 And T4 <= T2 And T4 <= T3) Then
MsgBox ("Assessment 2: The overall mission/task risk level before controls are
implemented is Extremely High (EH), and the average outcome for mission/task level
before controls are implemented is   ") & Value
End If

If (Ehigh > 0) Then MsgBox "Assessment 3: The overall risk is (EH) extremely high."

If (High > 0 And Ehigh = 0) Then MsgBox "Assessment 3: The overall risk is (H) High."
```

If (Med > 0 And Ehigh = 0 And High = 0) Then MsgBox "Assessment 3: The overall risk is (M) Moderate."

If (Low > 0 And Ehigh = 0 And High = 0 And Med = 0) Then MsgBox "Assessment 3: The overall risk is (L) Low."

End If
End Sub

All of the text box routines below are activated by the event by double clicking on the textbox. This event causes the input box to active. At this point a user type in a task name stored in the database. If the name is stored in the database, the information related to the task is stored in the respective combo columns for that particular row.

```
Private Sub TextBox1_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim DB1 As Database
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst
TextBox1.Value = .Fields("Task")
ComboBox61.Clear
ComboBox61.Value = "Use drop down button to make selections."
ComboBox82.Value = "Use drop down button to make selecitons."
ComboBox42.Value = "Use drop down button to make selections."

ComboBox61.AddItem .Fields("Hazard")
ComboBox82.Clear
ComboBox82.Value = "Use drop down button to make selections."
ComboBox82.AddItem .Fields("Dcontrols")
```

```
ComboBox42.Clear
ComboBox42.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox61.AddItem .Fields("Hazard")
ComboBox82.AddItem .Fields("Dcontrols")
ComboBox42.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox10_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox10.Value = .Fields("Task")
ComboBox70.Clear
ComboBox70.Value = "Use drop down button to make selectons."
ComboBox90.Value = "Use drop down button to make selectons."
ComboBox50.Value = "Use drop down button to make selectons."

ComboBox70.AddItem .Fields("Hazard")
ComboBox90.Clear
ComboBox90.AddItem .Fields("Dcontrols")
ComboBox50.Clear
```

Appendix D: Risk Management Worksheet

```
ComboBox50.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox70.AddItem .Fields("Hazard")
ComboBox90.AddItem .Fields("Dcontrols")
ComboBox50.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close

End Sub



Private Sub TextBox11_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox11.Value = .Fields("Task")
ComboBox71.Clear
ComboBox71.Value = "Use drop down button to make selectons."
ComboBox91.Value = "Use drop down button to make selectons."
ComboBox51.Value = "Use drop down button to make selectons."
ComboBox71.AddItem .Fields("Hazard")
ComboBox91.Clear
ComboBox91.AddItem .Fields("Dcontrols")
```

```
ComboBox51.Clear
ComboBox51.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox71.AddItem .Fields("Hazard")
ComboBox91.AddItem .Fields("Dcontrols")
ComboBox51.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub



Private Sub TextBox12_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox12.Value = .Fields("Task")
ComboBox72.Clear
ComboBox72.Value = "Use drop down button to make selectons."
ComboBox92.Value = "Use drop down button to make selectons."
ComboBox52.Value = "Use drop down button to make selectons."
ComboBox72.AddItem .Fields("Hazard")
ComboBox92.Clear
```

```
ComboBox92.AddItem .Fields("Dcontrols")
ComboBox52.Clear
ComboBox52.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox72.AddItem .Fields("Hazard")
ComboBox92.AddItem .Fields("Dcontrols")
ComboBox52.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox13_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox13.Value = .Fields("Task")
ComboBox73.Clear

ComboBox73.Value = "Use drop down button to make selectons."
ComboBox93.Value = "Use drop down button to make selectons."
ComboBox53.Value = "Use drop down button to make selectons."
ComboBox73.AddItem .Fields("Hazard")
ComboBox93.Clear
ComboBox93.AddItem .Fields("Dcontrols")
```

```
ComboBox53.Clear
ComboBox53.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox73.AddItem .Fields("Hazard")
ComboBox93.AddItem .Fields("Dcontrols")
ComboBox53.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub


Private Sub TextBox14_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox14.Value = .Fields("Task")
ComboBox74.Clear
ComboBox74.Value = "Use drop down button to make selectons."
ComboBox94.Value = "Use drop down button to make selectons."
ComboBox54.Value = "Use drop down button to make selectons."
ComboBox74.AddItem .Fields("Hazard")
ComboBox94.Clear
ComboBox94.AddItem .Fields("Dcontrols")
ComboBox54.Clear
```

```
ComboBox54.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox74.AddItem .Fields("Hazard")
ComboBox94.AddItem .Fields("Dcontrols")
ComboBox54.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox15_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox15.Value = .Fields("Task")
ComboBox75.Value = "Use drop down button to make selectons."
ComboBox95.Value = "Use drop down button to make se"
ComboBox55.Value = "Use drop down button to make se"
ComboBox75.Clear
ComboBox75.AddItem .Fields("Hazard")
ComboBox95.Clear
ComboBox95.AddItem .Fields("Dcontrols")
ComboBox55.Clear
ComboBox55.AddItem .Fields("Icontrols")
While Not .EOF
```

```
.MoveNext
ComboBox75.AddItem .Fields("Hazard")
ComboBox95.AddItem .Fields("Dcontrols")
ComboBox55.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox16_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox16.Value = .Fields("Task")
ComboBox76.Clear
ComboBox76.Value = "Use drop down button to make selectons."
ComboBox96.Value = "Use drop down button to make selectons."
ComboBox56.Value = "Use drop down button to make selectons."
ComboBox76.AddItem .Fields("Hazard")
ComboBox96.Clear
ComboBox96.AddItem .Fields("Dcontrols")
ComboBox56.Clear
ComboBox56.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox76.AddItem .Fields("Hazard")
```

```
ComboBox96.AddItem .Fields("Dcontrols")
ComboBox56.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox17_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox17.Value = .Fields("Task")
ComboBox77.Clear
ComboBox77.Value = "Use drop down button to make selectons."
ComboBox97.Value = "Use drop down button to make selectons."
ComboBox57.Value = "Use drop down button to make selectons."

ComboBox77.AddItem .Fields("Hazard")
ComboBox97.Clear
ComboBox97.AddItem .Fields("Dcontrols")
ComboBox57.Clear
ComboBox57.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox77.AddItem .Fields("Hazard")
ComboBox97.AddItem .Fields("Dcontrols")
```

```
ComboBox57.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox18_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox18.Value = .Fields("Task")
ComboBox78.Clear
ComboBox78.Value = "Use drop down button to make selectons."
ComboBox98.Value = "Use drop down button to make selectons."
ComboBox58.Value = "Use drop down button to make selectons."
ComboBox78.AddItem .Fields("Hazard")
ComboBox98.Clear
ComboBox98.AddItem .Fields("Dcontrols")
ComboBox58.Clear
ComboBox58.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox78.AddItem .Fields("Hazard")
ComboBox98.AddItem .Fields("Dcontrols")
ComboBox58.AddItem .Fields("Icontrols")
Wend
```

```
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox19_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If

QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox19.Value = .Fields("Task")
ComboBox61.Clear
ComboBox61.Value = "Use drop down button to make selectons."
ComboBox82.Value = "Use drop down button to make selectons."
ComboBox42.Value = "Use drop down button to make selectons."
ComboBox61.AddItem .Fields("Hazard")
ComboBox82.Clear
ComboBox82.AddItem .Fields("Dcontrols")
ComboBox42.Clear
ComboBox42.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox61.AddItem .Fields("Hazard")
ComboBox82.AddItem .Fields("Dcontrols")
ComboBox42.AddItem .Fields("Icontrols")
Wend
End With
```

```
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox2_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst
TextBox2.Value = .Fields("Task")
ComboBox62.Clear
ComboBox62.Value = "Use drop down button to make selectons."
ComboBox81.Value = "Use drop down button to make selectons."
ComboBox41.Value = "Use drop down button to make selectons."

ComboBox62.AddItem .Fields("Hazard")
ComboBox81.Clear
ComboBox81.AddItem .Fields("Dcontrols")
ComboBox41.Clear
ComboBox41.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox62.AddItem .Fields("Hazard")
ComboBox81.AddItem .Fields("Dcontrols")
ComboBox41.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
```

```
ErrorHandler:
DB1.Close
End Sub

Private Sub TextBox20_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst
TextBox20.Value = .Fields("Task")
ComboBox80.Clear
ComboBox80.Value = "Use drop down button to make selectons."
ComboBox100.Value = "Use drop down button to make selectons."
ComboBox60.Value = "Use drop down button to make selectons."
ComboBox80.AddItem .Fields("Hazard")
ComboBox100.Clear
ComboBox100.AddItem .Fields("Dcontrols")
ComboBox60.Clear
ComboBox60.AddItem .Fields("Icontrols")
'MsgBox "Customer's name is " & .Fields("Hazard") & "."
While Not .EOF
.MoveNext
ComboBox80.AddItem .Fields("Hazard")
ComboBox100.AddItem .Fields("Dcontrols")
ComboBox60.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
```

```
End Sub

Private Sub TextBox3_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
If TextBox3.Value <> " " Then
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox3.Value = .Fields("Task")
ComboBox63.Clear
ComboBox63.Value = "Use drop down button to make selectons."
ComboBox83.Value = "Use drop down button to make selectons."
ComboBox43.Value = "Use drop down button to make selectons."
ComboBox63.AddItem .Fields("Hazard")
ComboBox83.Clear
ComboBox83.AddItem .Fields("Dcontrols")
ComboBox43.Clear
ComboBox43.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox63.AddItem .Fields("Hazard")
ComboBox83.AddItem .Fields("Dcontrols")
ComboBox43.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End If
```

```
End Sub

Private Sub TextBox4_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox4.Value = .Fields("Task")
ComboBox64.Clear
ComboBox64.Value = "Use drop down button to make selectons."
ComboBox84.Value = "Use drop down button to make selectons."
ComboBox44.Value = "Use drop down button to make selectons."
ComboBox64.AddItem .Fields("Hazard")
ComboBox84.Clear
ComboBox84.AddItem .Fields("Dcontrols")
ComboBox44.Clear
ComboBox44.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox64.AddItem .Fields("Hazard")
ComboBox84.AddItem .Fields("Dcontrols")
ComboBox44.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close

End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub TextBox5_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox5.Value = .Fields("Task")
ComboBox65.Clear
ComboBox65.Value = "Use drop down button to make selectons."
ComboBox85.Value = "Use drop down button to make selectons."
ComboBox45.Value = "Use drop down button to make selectons."
ComboBox65.AddItem .Fields("Hazard")
ComboBox85.Clear
ComboBox85.AddItem .Fields("Dcontrols")
ComboBox45.Clear
ComboBox45.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox65.AddItem .Fields("Hazard")
ComboBox85.AddItem .Fields("Dcontrols")
ComboBox45.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub TextBox6_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If

QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox6.Value = .Fields("Task")
ComboBox66.Clear
ComboBox66.Value = "Use drop down button to make selectons."
ComboBox86.Value = "Use drop down button to make selectons."
ComboBox46.Value = "Use drop down button to make selectons."
ComboBox66.AddItem .Fields("Hazard")
ComboBox86.Clear
ComboBox86.AddItem .Fields("Dcontrols")
ComboBox46.Clear
ComboBox46.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox66.AddItem .Fields("Hazard")
ComboBox86.AddItem .Fields("Dcontrols")
ComboBox46.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub TextBox7_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox7.Value = .Fields("Task")
ComboBox67.Clear
ComboBox67.Value = "Use drop down button to make selectons."
ComboBox87.Value = "Use drop down button to make selectons."
ComboBox47.Value = "Use drop down button to make selectons."
ComboBox67.AddItem .Fields("Hazard")
ComboBox87.Clear
ComboBox87.AddItem .Fields("Dcontrols")
ComboBox47.Clear
ComboBox47.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox67.AddItem .Fields("Hazard")
ComboBox87.AddItem .Fields("Dcontrols")
ComboBox47.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close

End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub TextBox8_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox8.Value = .Fields("Task")
ComboBox68.Clear
ComboBox68.Value = "Use drop down button to make selectons."
ComboBox88.Value = "Use drop down button to make selectons."
ComboBox48.Value = "Use drop down button to make selectons."
ComboBox68.AddItem .Fields("Hazard")
ComboBox88.Clear
ComboBox88.AddItem .Fields("Dcontrols")
ComboBox48.Clear
ComboBox48.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox68.AddItem .Fields("Hazard")
ComboBox88.AddItem .Fields("Dcontrols")
ComboBox48.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub
```

Appendix D: Risk Management Worksheet

```
Private Sub TextBox9_DblClick(ByVal Cancel As MSForms.ReturnBoolean)
Dim RS1 As Recordset
Dim QRY1 As QueryDef
Dim QueryString As String
Dim temp As String
On Error GoTo ErrorHandler
QueryString = "PARAMETERS [Param1]TEXT; " & _
"SELECT * FROM Risk WHERE [Task] = [Param1]"
Set DB1 = OpenDatabase(ThisWorkbook.Path & "\Wilson.mdb")
If DB1.QueryDefs.Count < 1 Then
Set QRY1 = DB1.CreateQueryDef("Query1", QueryString)
Else
Set QRY1 = DB1.QueryDefs("Query1")
QRY1.Sql = QueryString
End If
QRY1.Parameters("Param1") = InputBox("Enter task name.")
Set RS1 = QRY1.OpenRecordset(dbOpenDynaset)
With RS1
.MoveFirst

TextBox9.Value = .Fields("Task")
ComboBox69.Clear
ComboBox69.Value = "Use drop down button to make selectons."
ComboBox89.Value = "Use drop down button to make selectons."
ComboBox49.Value = "Use drop down button to make selectons."
ComboBox69.AddItem .Fields("Hazard")
ComboBox89.Clear
ComboBox89.AddItem .Fields("Dcontrols")
ComboBox49.Clear
ComboBox49.AddItem .Fields("Icontrols")
While Not .EOF
.MoveNext
ComboBox69.AddItem .Fields("Hazard")
ComboBox89.AddItem .Fields("Dcontrols")
ComboBox49.AddItem .Fields("Icontrols")
Wend
End With
DB1.Close
Exit Sub
ErrorHandler:
DB1.Close
End Sub
```

Appendix E: Storage Problem

This routine adds the optimal solution configuration to the Excel Workbook page

```
Sub word()
Workbooks.OpenText FileName:="C:\My Documents\CUT.DOC", Origin:=xlWindows,
_
StartRow:=1, DataType:=xlDelimited, TextQualifier:=xlDoubleQuote, _
ConsecutiveDelimiter:=False, Tab:=True, Semicolon:=False, Comma:=False _
, Space:=False, Other:=False, FieldInfo:=Array(1, 1)
Sheets("CUT").Select
Sheets("CUT").Move Before:=Workbooks("Storage6.xls").Sheets(2)
End Sub
```

This routine puts the cursor at the starting text on the worksheet solution page.

```
Sub optimal()
Cells.Find(What:="optimal configuration Pattern", After:=ActiveCell, LookIn:= _
xlFormulas, LookAt:=xlPart, SearchOrder:=xlByRows, SearchDirection:= _
xlNext, MatchCase:=False).Activate
End Sub
```

This routine configures the boxes and finds the smallest "foot print" before the problem is optimized.

```
Sub sort1()
For i = 3 To 52
Range(Worksheets("Storage").Cells(i, 2), Worksheets("Storage").Cells(i, 4)).Select
Selection.sort Key1:=Worksheets("Storage").Cells(i, 2), Order1:=xlDescending,
Header:=xlGuess, _
OrderCustom:=1, MatchCase:=False, Orientation:=xlLeftToRight
Next i
Range("A1").Select
End Sub
```

This routine solves the 2 dimensional cutting stock problem.

```
Option Base 1
Sub Find()
With Application.WorksheetFunction
Dim TA As Long
Const TOL1 As Single = 0.0001
Const TOL2 As Single = 0.00001
Dim NROW As Long
Dim IPR As Long
Dim NR As Long
Dim SL As Long
```

Appendix E: Storage Problem

```
Dim SW As Long
Dim DX As Long
Dim DY As Long
Dim ML As Long
Dim MW As Long
Dim M1 As Long
Dim M2 As Long
Dim IT As Long
Dim ED As Long
Dim SC As Single
Dim AC As Single
Dim ZB As Single
Dim IX As Long

1040  NROW = Worksheets("Storage").Cells(20, 6)
1080  IPR = 2
1110   SL = Worksheets("Storage").Cells(9, 6)
1115   SW = Worksheets("Storage").Cells(13, 6)
1116   SC = 1
1130  TA = 5000 * NROW
Dim OL() As Long
ReDim OL(NROW)
Dim OW() As Long
ReDim OW(NROW)
Dim X() As Long
ReDim X(NROW)
Dim BL() As Long
ReDim BL(NROW)
Dim BW() As Long
ReDim BW(NROW)
Dim OQ() As Variant
ReDim OQ(NROW)
Dim T1() As Long
ReDim T1(TA)
Dim T2() As Long
ReDim T2(TA)
Dim TX() As Long
ReDim TX(TA)
Dim TY() As Long
ReDim TY(TA)
Dim R1() As Long
ReDim R1(TA)
Dim R2() As Long
ReDim R2(TA)
Dim BBAR() As Variant
ReDim BBAR(NROW)
```

Appendix E: Storage Problem

```
Dim F() As Variant
ReDim F(NROW)
Dim AR() As Variant
ReDim AR(NROW)
Dim BA() As Variant
ReDim BA(NROW)
Dim CB() As Variant
ReDim CB(NROW)
Dim PI() As Variant
ReDim PI(NROW)
Dim BE() As Variant
ReDim BE(NROW)
Dim RT() As Variant
ReDim RT(NROW)
Dim PA() As Long
ReDim PA(NROW, NROW)
Dim W() As Long
ReDim W(0 To SL, 0 To SW)
Dim L() As Long
ReDim L(0 To SL, 0 To SW)
Dim BI() As Variant
ReDim BI(NROW, NROW)
Dim V1() As Variant
ReDim V1(SL, SW)
Dim name3(50) As String
Dim SWI As String
Dim fp As String

1225 For i = 1 To NROW
1226 OL(i) = Worksheets("Storage").Cells(i + 2, 2)
1227 OW(i) = Worksheets("Storage").Cells(i + 2, 3)
1228 OQ(i) = Worksheets("Storage").Cells(i + 2, 5)
1229Next I

1230 If IPR = 1 Then GoTo 1500
1235 Open "CUT.DOC" For Output As #1
1240 Print #1,: Print #1, "***  INPUT  DATA  ***": Print #1,
1250 Print #1, "* STOCK LENGTH ,STOCK WIDTH,STOCK COST *": Print #1,
1260 Print #1, SL, SW, SC
1270 Print #1,
1280 Print #1,
1290 Print #1, "* ORDER LENGTH , ORDER WIDTH , ORDER QUANTITY *"
1295 Print #1,
1300 For j = 1 To NROW
1310   Print #1, OL(j), OW(j), OQ(j)
1320 Next j: Print #1,
```

Appendix E: Storage Problem

```
1500 Rem ********
1510 Rem  INISOL
1520 Rem ********
1530 IT = 1
1540 If IPR = 2 Then GoTo 1560
1560 Print #1, "***** ITERATION"; IT; "*****"
1570 Print #1,
1580 For i = 1 To NROW
1590   For j = 1 To NROW
1600     If i <> j Then PA(i, j) = 0: GoTo 1620
1610     PA(i, j) = Int(SL / OL(i)) * Int(SW / OW(i))
1620   Next j
1630 Next i
1640 For i = 1 To NROW: BL(i) = SL: BW(i) = SW: Next i
1650 For i = 1 To NROW: CB(i) = SC: Next i
1660 AC = SC: ML = SL: MW = SW
1680 For i = 1 To NROW
1690   If ML <= OL(i) Then GoTo 1710
1700   ML = OL(i)
1710 Next i
1720 For i = 1 To NROW
1730   If MW <= OW(i) Then GoTo 1750
1740   MW = OW(i)
1750 Next i
1760 For i = 1 To NROW: For j = 1 To NROW
1770   If i = j Then BI(i, j) = 1 / PA(i, j): GoTo 1790
1780   BI(i, j) = 0
1790 Next j: Next i
1800 For i = 1 To NROW
1810   BBAR(i) = 0
1820   For j = 1 To NROW
1830     BBAR(i) = BI(i, j) * OQ(j) + BBAR(i)
1840   Next j
1850 Next i
1860 ZB = 0
1870 For i = 1 To NROW
1880   ZB = CB(i) * BBAR(i) + ZB
1890 Next i
1900 If IPR = 1 Then GoTo 2500
1910 Print #1,: Print #1, "** INITIAL CUTTING Pattern **": Print #1,
1920 For i = 1 To NROW
1930   Print #1, "Pattern ("; i; ")"
1940   Print #1, "STOCK RECTANGLE="; BL(i); "*"; BW(i); ""; Spc(3);
"QUANTITY="; BBAR(i)
1950   For j = 1 To NROW
```

Appendix E: Storage Problem

```
1960       If PA(j, i) = 0 Then GoTo 1990
1970       Print #1, "ORDER RECTANGLE="; OL(i); "*"; OW(i); ""; Spc(3);
1980       Print #1, "number of items"; PA(j, i)
1990     Next j
2000     Print #1,
2010  Next i
2020  Print #1,
2030  Print #1, "Minimum number of shelves required is="; ZB
2040  Print #1,

2500  Rem   ******
2510  Rem    MAIN
2520  Rem   ******
2530  GoSub 3000
2540  GoSub 3500
2550  GoSub 5500
2560  GoSub 6000
2570  GoSub 6500
2580  GoSub 7000
2590  IT = IT + 1
2600  If IPR = 1 Then GoTo 2630
2610  Print #1,: Print #1, "**** ITERATION "; IT; " ****": Print #1,
2620  Print #1,: Print #1,
2625  'Print "***  ITERATION"; IT; "  ***"
2630  GoTo 2530

3000  Rem  **********************
3010  Rem   BTRAN:CALCULATING PI
3020  Rem  **********************
3030  For i = 1 To NROW
3040    PI(i) = 0
3050    For j = 1 To NROW
3060      PI(i) = CB(j) * BI(j, i) + PI(i)
3070    Next j
3080  Next i
3090  If IPR = 1 Then GoTo 3130
3100  Print #1, "** BTRAN: CALCULATING PI **": Print #1,
3110  For i = 1 To NROW: Print #1, PI(i),: Next i
3120  Print #1,
3130  Return

3500  Rem  ******************
3510  Rem   ENTERING COLUMN
3520  Rem  *****************
3530  If IPR = 1 Then GoTo 3550
3540  Print #1,
```

Appendix E: Storage Problem

```
3550 For i = 1 To NROW
3560   X(i) = 0
3570 Next i
3580 Rem SLACK ENTERING
3590 SC = AC
3600 For i = 1 To NROW
3610   If PI(i) < -TOL1 Then X(i) = -1: SC = 0: GoTo 5080
3620 Next i
3630 If IPR = 1 Then GoTo 3650
3650 For i = 1 To SL: For j = 1 To SW
3660   V1(i, j) = 0: L(i, j) = 0: W(i, j) = 0
3670 Next j: Next i
3680 For K = 1 To NROW
3690   If PI(K) < TOL1 Then GoTo 3770
3700   For i = 1 To SL
3710     For j = 1 To SW
3720       If i >= OL(K) And j >= OW(K) Then V2 = PI(K): GoTo 3740
3730       GoTo 3750
3740       If V1(i, j) < V2 Then V1(i, j) = V2
3750     Next j
3760   Next i
3770 Next K
3780 For K = 1 To NROW
3790   If PI(K) < TOL1 Then GoTo 3820
3800   V1(OL(K), OW(K)) = PI(K)
3810   L(OL(K), OW(K)) = OL(K): W(OL(K), OW(K)) = OW(K)
3820 Next K
3830 L(0, 0) = 0: W(0, 0) = 0
3840 XX = 1: YY = 1
3850 Rem HORIZONTAL
3860 IX = 1
3865 NOX = 0
3870 If IX + XX <= SL Then GoTo 3890
3880 GoTo 3945
3890 V = V1(IX, YY) + V1(XX, YY)
3900 If V - V1(IX + XX, YY) > TOL2 Then GoTo 3930
3910 If V1(IX + XX, YY) - V > TOL2 Then GoTo 3940
3920 If (V1(IX, YY) * V1(XX, YY) <> 0) Then L(IX + XX, YY) = IX: W(IX + XX,
YY) = YY: GoTo 3940
3930 V1(IX + XX, YY) = V: L(IX + XX, YY) = IX: W(IX + XX, YY) = YY
3940 If IX < XX Then IX = IX + 1: NOX = 1: GoTo 3870
3945 If NOX = 1 Then GoTo 3865
3950 Rem VERTICAL
3960 Y = 1
3965 NOY = 0
3970 If Y + YY <= SW Then GoTo 3990
```

Appendix E: Storage Problem

```
3980  GoTo 4045
3990  V = V1(XX, Y) + V1(XX, YY)
4000  If V - V1(XX, Y + YY) > TOL2 Then GoTo 4030
4010  If V1(XX, Y + YY) - V > TOL2 Then GoTo 4040
4020  If (V1(XX, Y) * V1(XX, YY) <> 0) Then L(XX, Y + YY) = XX: W(XX, Y + YY)
= Y: GoTo 4040
4030  V1(XX, Y + YY) = V: W(XX, Y + YY) = Y: L(XX, Y + YY) = XX
4040  If Y < YY Then Y = Y + 1: NOY = 1: GoTo 3970
4045  If NOY = 1 Then GoTo 3965
4050  If XX < SL Then XX = XX + 1: GoTo 3850
4060  If YY < SW Then YY = YY + 1: XX = 1: GoTo 3850
4070  Rem BACKTRACKING
4080  NL = SL: NW = SW
4090  If L(NL, NW) * W(NL, NW) <> 0 Then GoTo 4120
4100  If L(NL, NW) = 0 Then NW = NW - 1: GoTo 4090
4110  If W(NL, NW) = 0 Then NL = NL - 1: GoTo 4090
4120  AX = NL: AY = NW
4130  M1 = 0
4140  If L(AX, AY) <> AX And W(AX, AY) <> AY Then GoTo 4280
4150  If W(AX, AY) <> AY Then GoTo 4220
4160  Rem W(AX,AY)=AY
4170  DX = L(AX, AY): DY = W(AX, AY)
4180  GoSub 4380
4190  DX = AX - L(AX, AY): DY = W(AX, AY)
4200  GoSub 4380
4210  GoTo 4370
4220  Rem L(AX,AY)=AX
4230  DX = L(AX, AY): DY = W(AX, AY)
4240  GoSub 4380
4250  DX = L(AX, AY): DY = AY - W(AX, AY)
4260  GoSub 4380
4270  GoTo 4370
4280  Rem L(AX,AY)<>AX AND W(AX,AY)<>AY
4290  DX = L(AX, AY): DY = W(AX, AY)
4300  GoSub 4380
4310  DX = AX - L(AX, AY): DY = W(AX, AY)
4320  GoSub 4380
4330  DX = L(AX, AY): DY = AY - W(AX, AY)
4340  GoSub 4380
4350  DX = AX - L(AX, AY): DY = AY - W(AX, AY)
4360  GoSub 4380
4370  GoTo 4460
4380  Rem
4390  If DX < ML Or DY < MW Then GoTo 4450
4395  K1 = 1
4400  If L(DX, DY) * W(DX, DY) <> 0 Then GoTo 4430
```

Appendix E: Storage Problem

```
4403 i = DX: j = DY:
4406 If ((V1(DX, DY) - V1(DX - K1, DY) <= TOL1) And (V1(DX, DY) - V1(DX - K1,
DY) >= -TOL1)) Then DX = DX - K1: GoTo 4425
4410 If ((V1(DX, DY) - V1(DX, DY - K1) <= TOL1) And (V1(DX, DY) - V1(DX, DY
- K1) >= -TOL1)) Then DY = DY - K1: GoTo 4425
4415 If ((V1(DX, DY) - V1(DX - K1, DY - K1) <= TOL1) And (V1(DX, DY) - V1(DX
- K1, DY - K1) >= -TOL1)) Then DX = DX - K1: DY = DY - K1: GoTo 4425
4420 If ((i = DX) And (j = DY)) Then K1 = K1 + 1
4425 GoTo 4400
4430 M1 = M1 + 1
4440 TX(M1) = DX: TY(M1) = DY
4450 Return
4460 Rem
4470 K = 0: M2 = 0
4480 For i = 1 To M1
4490   If L(TX(i), TY(i)) <> TX(i) Or W(TX(i), TY(i)) <> TY(i) Then GoTo 4540
4500   Rem L(TX(I),TY(I))=TX(I):W(TX(I),TY(I))=TY(I)
4510   K = K + 1
4520   R1(K) = TX(i): R2(K) = TY(i)
4530   GoTo 4770
4540   If L(TX(i), TY(i)) <> TX(i) Then GoTo 4610
4550 Rem L(TX(I),TY(I))=TX(I)
4560   DX = L(TX(i), TY(i)): DY = W(TX(i), TY(i))
4570   GoSub 4790
4580   DX = L(TX(i), TY(i)): DY = TY(i) - W(TX(i), TY(i))
4590   GoSub 4790
4600   GoTo 4770
4610   If W(TX(i), TY(i)) <> TY(i) Then GoTo 4680
4620 Rem W(TX(I),TY(I))=TY(I)
4630   DX = L(TX(i), TY(i)): DY = W(TX(i), TY(i))
4640   GoSub 4790
4650   DX = TX(i) - L(TX(i), TY(i)): DY = W(TX(i), TY(i))
4660   GoSub 4790
4670   GoTo 4770
4680 Rem L(TX(I),TY(I))<>TX(I) AND W(TX(I),TY(I))<>TY(I)
4690   DX = L(TX(i), TY(i)): DY = W(TX(i), TY(i))
4700   GoSub 4790
4710   DX = TX(i) - L(TX(i), TY(i)): DY = W(TX(i), TY(i))
4720   GoSub 4790
4730   DX = L(TX(i), TY(i)): DY = TY(i) - W(TX(i), TY(i))
4740   GoSub 4790
4750   DX = TX(i) - L(TX(i), TY(i)): DY = TY(i) - W(TX(i), TY(i))
4760   GoSub 4790
4770 Next i
4780 GoTo 4870
4790 Rem
```

Appendix E: Storage Problem

```
4800 If DX < ML Or DY < MW Then GoTo 4860
4805 K1 = 1
4810 If L(DX, DY) * W(DX, DY) <> 0 Then GoTo 4840
4813 i = DX: j = DY:
4816 If ((V1(DX, DY) - V1(DX - K1, DY) <= TOL1) And (V1(DX, DY) - V1(DX - K1, DY) >= -TOL1)) Then DX = DX - K1: GoTo 4835
4820 If ((V1(DX, DY) - V1(DX, DY - K1) <= TOL1) And (V1(DX, DY) - V1(DX, DY - K1) >= -TOL1)) Then DY = DY - K1: GoTo 4835
4825 If ((V1(DX, DY) - V1(DX - K1, DY - K1) <= TOL1) And (V1(DX, DY) - V1(DX - K1, DY - K1) >= -TOL1)) Then DX = DX - K1: DY = DY - K1: GoTo 4835
4830 If ((i = DX) And (j = DY)) Then K1 = K1 + 1
4835 GoTo 4810
4840 M2 = M2 + 1
4850 T1(M2) = DX: T2(M2) = DY
4860 Return
4870 Rem
4880 If M2 = 0 Then GoTo 4950
4890 For i = 1 To M2
4900   TX(i) = T1(i): TY(i) = T2(i)
4910 Next i
4920 M1 = M2
4930 M2 = 0: GoTo 4480
4940 Rem ENTERING COLUMN
4950 For i = 1 To NROW: X(i) = 0: Next i
4960 For j = 1 To K
4970   For i = 1 To NROW
4980     If PI(i) < TOL1 Then GoTo 5000
4990     If R1(j) = OL(i) And R2(j) = OW(i) Then X(i) = X(i) + 1
5000   Next i
5010 Next j
5020 Z1 = 0
5030 For i = 1 To NROW
5040   If PI(i) = 0 Then GoTo 5060
5050   Z1 = X(i) * PI(i) + Z1
5060 Next i
5070 If SC - Z1 > -TOL1 Then GoTo 7500
5080 If IPR = 1 Then GoTo 5130
5090 Print #1,: Print #1,
5100 Print #1, "** ENTERING COLUMN I=1,NROW **": Print #1,
5110 For i = 1 To NROW: Print #1, X(i),: Next i
5120 Print #1,: Print #1,
5130 Return

5500 Rem ******************
5510 Rem  FTRAN:UPDATE COL
5520 Rem ******************
```

Appendix E: Storage Problem

```
5530 For i = 1 To NROW
5540   BA(i) = 0
5550   For j = 1 To NROW
5560     BA(i) = BI(i, j) * X(j) + BA(i)
5570   Next j
5580 Next i
5590 For i = 1 To NROW
5600   AR(i) = BA(i)
5610 Next i
5620 If IPR = 1 Then GoTo 5650
5630 Print #1,: Print #1, "** FTRAN: UPDATE COL **": Print #1,
5640 For i = 1 To NROW: Print #1, AR(i),: Next i
5650 Return

6000 Rem  *******
6010 Rem  CHUZR
6020 Rem  *******
6030 MINR = 10000000000#
6040 For i = 1 To NROW
6050   RT(i) = 10000000000#
6060   If AR(i) <= 0 Then GoTo 6080
6070   RT(i) = BBAR(i) / AR(i)
6080 Next i
6090 For i = 1 To NROW
6100   If MINR <= RT(i) Then GoTo 6120
6110   MINR = RT(i): NR = i
6120 Next i
6130 BL(NR) = SL: BW(NR) = SW: CB(NR) = SC
6160 For i = 1 To NROW
6170   PA(i, NR) = X(i)
6180 Next i
6190 If IPR = 1 Then GoTo 6250
6200 Print #1,
6210 Print #1,: Print #1, "** CHUZR:PIVOT ROW **": Print #1,
6220 Print #1,: Print #1, "BL("; NR; ")"; "*"; "BW("; NR; ")", "CB("; NR; ")", "RT(";
NR; ")"
6230 Print #1,
6240 Print #1, BL(NR); "*"; BW(NR), CB(NR), RT(NR)
6250 Return

6500 Rem  ******
6510 Rem  BINV
6520 Rem  ******
6530 TMP = AR(NR)
6540 For i = 1 To NROW
6550   F(i) = -1 * AR(i) / TMP
```

Appendix E: Storage Problem

```
6560 Next i
6570 F(NR) = 1 / TMP
6580 For i = 1 To NROW
6590   If i = NR Then GoTo 6630
6600   For j = 1 To NROW
6610     BI(i, j) = BI(i, j) + F(i) * BI(NR, j)
6620   Next j
6630 Next i
6640 For j = 1 To NROW
6650   BI(NR, j) = F(NR) * BI(NR, j)
6660 Next j
6670 If IPR = 1 Then GoTo 6710
6680 Print #1,: Print #1, "** UPBINV: UPDATE B INVERSE **": Print #1,
6690 Print #1, "BI(I,J),I=1,NROW,J=1,NROW  = ": Print #1,
6700 For i = 1 To NROW: For j = 1 To NROW: Print #1, BI(i, j),: Next j: Next i
6710 Return

7000 Rem  *****************
7010 Rem  UPDATE SOLUTION
7020 Rem  *****************
7030 For i = 1 To NROW
7040   BBAR(i) = 0
7050   For j = 1 To NROW
7060     BBAR(i) = BI(i, j) * OQ(j) + BBAR(i)
7070 Next j: Next i
7080 ZB = 0
7090 For i = 1 To NROW
7100   ZB = CB(i) * BBAR(i) + ZB
7110 Next i
7120 If IPR = 1 Then GoTo 7260
7130 Print #1,: Print #1, "** UPSOL: UPDATED CUTTING Pattern **": Print #1,
7140 For i = 1 To NROW
7150   If BBAR(i) = 0 Then GoTo 7240
7160   Print #1, "Pattern ("; i; ")"
7170   Print #1, "Storage rectangle "; BL(i); "*"; BW(i); ""; Spc(3);: Print #1,
"QUANTITY="; BBAR(i)
7180   For j = 1 To NROW
7190     If PA(j, i) = 0 Then GoTo 7220
7200     Print #1, "Item Rectangle "; OL(j); "*"; OW(j); ""; Spc(3);
7210     Print #1, "number of items"; PA(j, i)
7220   Next j
7230   Print #1,
7240 Next i
7250 Print #1,: Print #1, "Minimum number of shelves required is="; ZB: Print #1,
7260 Return
```

Appendix E: Storage Problem

```
7500 Rem ********
7510 Rem  OPTSOL
7520 Rem ********
7530 Print #1,: Print #1,
7540 Print #1, "**** Optimal configuration Pattern ****": Print #1,
7550 For i = 1 To NROW
7560   If BBAR(i) = 0 Then GoTo 7650
7570   Print #1, "Pattern ("; i; ")"
7580   Print #1, "Storage rectangle "; BL(i); "*"; BW(i); ""; Spc(3);: Print #1,
"QUANTITY="; .RoundUp(BBAR(i), 0)
7590   For j = 1 To NROW
7600     If PA(j, i) = 0 Then GoTo 7630
7610     Print #1, "Item Rectangle "; OL(j); "*"; OW(j); ""; Spc(3);
7620     Print #1, "number of items"; PA(j, i)
7630   Next j
7640   Print #1,
7650 Next i
7660 ZB = 0
7670 For i = 1 To NROW
7680   ZB = CB(i) * BBAR(i) + ZB
7690 Next i
7700 Print #1, "Minimum number of shelves required is "; .RoundUp(ZB, 0)
8000 Close #1

End With
End Sub
```

REFERENCES

Anderson, D. R., Sweeney, D. J., Williams, T. A. (1994) An Introduction to Management Science: Quantitative Approaches to Decision Making, West Publishing Company, New York

Banks, J., Carson, J. S., Nelson, B. L. (1995) Introduction to Discrete-Event System Simulation, Prentice Hall, New Jersey

Center for Army Lessons Learned Web Page (CALL), http://call.army.mil.

Considinet, T., Personal Visit, 101st Air Assault Division G5, Fort Campbell, Kentucky, 17 April 1998

Cutting Stock problem, www.mcs.anl.gov/otc/Guide/CaseStudies/cutting /formulation.html

Dantzig, G. B. (1963) Linear Programming and Extensions, Princeton University Press, New Jersey

David, M. (1993) "The Role of Operations Research and Systems Analysis in Tomorrows's Army", Phalanx, 26(2):31-32 (June 1993).

Department of the Army. Combat Service Support. Field Manual 71-123 Washington: Chief of Staff, 30 September 1992.

Department of the Army. Risk Management. FM 100-14, Washington: Chief of Staff, 27 July 1995.

Henry, D., Personal Visit, Assistant S4, 2nd Brigade, Fort Stewart, Georgia, 22 September 1998

Hinson, J. , Personal Visit, Assistant G4, 3rd Infantry Division, Fort Stewart, Georgia, 23 September 1998

Jensen, P. A. (1987) Integer Programming Notes

Kim Chang-on, Sung Myung-Kee, and Ju Hien-Taek, Seoul National University, http://orlab.snu.ac.kr/

Klimack, W. Personal Visit. CDR, 2nd Battalion, 28th Infantry Regiment, Fort Jackson, South Carolina, 11 March 1998

McDonald, P. , Personal Visit, Assistant S3, 123rd Signal Battalion, Fort Stewart, Georgia, 24-25 September 1998

Combined Arms Support Page, http://www.cascom.army.mil/multi/operations_logistics_planner

Ramsey,R., Personal Visit, HHC 24th CSG, Fort Stewart, Georgia, 24 September 1998

Roach, F., Personal Visit, Assistant G4, 3rd Infantry Division, Fort Stewart, Georgia, 22 September 1998

Savage, S. L. (1998) Insight.xla: Business analysis Software, Duxbury Press, California

Shadrick, M., Personal Visit, Assistant G4, 3rd Infantry Division, Fort Stewart, Georgia, 23 September 1998

Sullivan, M. Personal Visit, 101st Air Assault Division G4, Fort Campbell, Kentucky, 17 April 1998

Wells, E., Harshbarger, S. (1997) Microsoft Excel 97: Developer's Handbook, Microsoft Press, Washington

Winston, W. L. (1994) Operations Research: Applications and Algorithms, Duxbury Press, California

Young, A. , Personal Visit, Assistant G4, 3rd Infantry Division, Fort Stewart, Georgia, 24 September 1998

Vita

CPT Blane Christopher Wilson was born at Fort, Benning, Georgia, 13 December 1966. He graduated from Kendrick High School in 1984. After high school, he attended Columbus State University, Columbus, Georgia. On August 1988, he graduated from Columbus State University, accepting a commission in the United States Army and earning a Bachelor of Science in Computer Science.

Afterwards, he attended the Signal Officer Basic Course, Fort Gordon, Georgia, and was assigned to the 22d Signal Brigade in Frankfurt, Germany. After his tour in Germany, he completed the Signal Officer Advanced Course, with a follow on assignment to the 24th Infantry Division, Fort Stewart, Georgia. During his tour, he was assigned as a Brigade Signal Officer and a Signal Company Commander.

CPT Wilson entered the Operations Analysis program at the Air Force Institute of Technology in January 1997. He and Winefreda have one child, Blane, Jr. After completion of his degree in operations analysis, CPT Wilson will be assigned to the Training Analysis Center (TRAC), Fort Leavenworth, Kansas.